REPORT DOCUMENTATION PAGE

Form Approved
OMB NO. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	October 29, 2007	Final Report from Sep 20, 2006 to Sep 19, 2007

4. TITLE AND SUBTITLE

Formal-language-theoretic Control & Coordination of Mobile Robots

5. FUNDING NUMBERS

W911NF-06-01-0469

6. AUTHOR(S)

Dr. Asok Ray, Principal Investigator

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

College of Engineering
Pennsylvania State University, University Park, PA 16802

8. PERFORMING ORGANIZATION REPORT NUMBER

FR CSF07001

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U. S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

50977.1-MA

11. SUPPLEMENTARY NOTES

The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.

12 a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12 b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

Objectives:
The goal of this research project is to enhance control and coordination of heterogeneous robots that include both Unmanned Ground Vehicles (UGVs) and Unmanned Aerial Vehicles (UAVs), supported by communication of onboard sensor and ancillary information among individual platforms and human users..

Results:
The research has formulated and experimentally validated robust adaptive algorithms and software codes for decision & control of mobile robotic platforms, as applied to real-time computation and execution of combat mission strategies. These algorithms are executable within a general-purpose programming language environment and make use of the generative power of formal-language-theoretic models instead of ad-hoc rule-based expert systems.

Future Plans:
- Formulation of operational intelligence models in a formal-language-theoretic setting
- Algorithm development for intelligent coordination of autonomous agent teams to accomplish complex mission tasks

14. SUBJECT TERMS

Formal language, automata theory, robotics, robotic languages, supervisory control, and risk analysis.

15. NUMBER OF PAGES

56

16. PRICE CODE

17. SECURITY CLASSIFICATION OR REPORT	18. SECURITY CLASSIFICATION ON THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	**UNCLASSIFIED**	**UNCLASSIFIED**	UU

NSN 7540-01-280-5500

Standard Form 298 (Rev.2-89)
Prescribed by ANSI Std. 239-18
298-102

Enclosure 1

MASTER COPY: PLEASE KEEP THIS "MEMORANDUM OF TRANSMITTAL" BLANK FOR REPRODUCTION PURPOSES. WHEN REPORTS ARE GENERATED UNDER THE ARO SPONSORSHIP, FORWARD A COMPLETED COPY OF THIS FORM WITH EACH REPORT SHIPMENT TO THE ARO. THIS WILL ASSURE PROPER IDENTIFICATION. NOT TO BE USED FOR INTERIM PROGRESS REPORTS; SEE PAGE 2 FOR INTERIM PROGRESS REPORT INSTRUCTIONS.

MEMORANDUM OF TRANSMITTAL

U.S. Army Research Office
ATTN: AMSRL-RO-BI (TR)
P.O. Box 12211
Research Triangle Park, NC 27709-2211

- ☐ Reprint (Orig + 2 copies)
- ☐ Manuscript (1 copy)
- ☐ Technical Report (Orig + 2 copies)
- ☑ Final Progress Report (Orig + 2 copies)
- ☐ Related Materials, Abstracts, Theses (1 copy)

CONTRACT/GRANT NUMBER: W911NF-06-01-0469

REPORT TITLE: Formal-language-theoretic Control and Coordination of Mobile Robots

is forwarded for your information.

SUBMITTED FOR PUBLICATION TO (applicable only if report is manuscript):

Sincerely,

Dr. Asok Ray, Principal Investigator

PROJECT:	Formal-language-theoretic Control & Coordination of Mobile Robots
GRANT NO.	W911NF-06-01-0469
REPORT TITLE:	Final Report on the 2006-2007 Robotic Language Project
PRINCIPAL INVESTIGATOR:	*Dr. Asok Ray*
OTHER KEY CONTRIBUTORS:	*Dr. Ishanu Chattopadhyay*
REPORT NUMBER:	TR 07002
STATUS:	Final Report for the Year 2006-2007
DATE:	October 29, 2007

FOREWORD

The project (Grant No. W911NF-06-01-0469) titled ***Formal-language-theoretic Control & Co-ordination of Mobile Robots*** was started on September 20, 2006 and ended on September 19, 2007. It was conducted under the leadership of Prof. Asok Ray and Dr. Ishanu Chattopadhyay, Pennsylvania State University, University Park, PA. This research project has developed a novel approach to control of co-operating and non-co-operating teams of autonomous and semi-autonomous agents and has made fundamental contributions to the enhancement of the state of the art in the field of Probabilistic Robotics. On the experimental side of the project, Penn State has developed the Networked Robotics and Sensors Laboratory (NRSL) for conducting research in robotics, which has been supported through a DURIP equipment grant.

The research work, conducted under this project, has resulted in 8 scholarly publications. This project has laid the groundwork for effectively transferring the newly developed technology to defense industry through future SBIR and STTR projects in collaboration with industry.

TABLE OF CONTENTS

STATEMENT OF THE PROBLEM STUDIED

The Future Combat Systems (FCS) program of the U.S. Army calls for collaboration among heterogeneous groups of semi-autonomous and autonomous mobile platforms, such as Unmanned Ground Vehicles (UGVs) and Unmanned Aerial Vehicles (UAVs), supported by communication of onboard sensor and ancillary information among individual platforms and human users. The research conducted under this project addresses the challenge of developing C^4I systems for applications to groups of autonomous vehicles that must adapt themselves to operate in uncertain and dynamically evolving environments of battlefields. The research has formulated and experimentally validated robust adaptive algorithms and software codes for decision & control of mobile robotic platforms, as applied to real-time computation and execution of combat mission strategies. These algorithms are executable within a general-purpose programming language environment and make use of the generative power of formal-language-theoretic models instead of ad-hoc rule-based expert systems. Research efforts have been concentrated in the following inter-related areas.

- Formulation of operational intelligence models in a formal-language-theoretic setting for: (i) autonomous agents and decision & control architectures, and (ii) interchangeable (i.e., platform-independent) human-machine interactions

- Algorithm development for intelligent coordination of autonomous agent teams to accomplish complex mission tasks with automated resource and risk optimization with (possibly) incomplete knowledge of work-space parameters due to: (i) insufficient information, and (ii) sensor and/or communication link failures

- Experimental validation of the mathematical tools and software codes in: (i) Systems Simulation Laboratory, and (ii) Networked Robotics & Sensors Laboratory, developed at Penn State under ARO-funded MURI and DURIP grants (PM: Dr. M-H. Chang)

SUMMARY OF THE MOST IMPORTANT RESULTS

Formal-language-theoretic Control & Co-ordination of Mobile Robots

Principal Investigator: Professor Asok Ray, Pennsylvania State University

Key Contributor: Dr. Ishanu Chattopadhyay, Pennsylvania State University

The verifiable outcomes of the project include novel algorithms, associated numerical methods, and validated software codes for decision & control of a group of mobile robots, focusing on the following issues.

- Coordination of heterogeneous groups of autonomous mobile platforms
- Robustness of controlled behavior under disturbances and loss of information
- Computational speed and memory requirements for real-time on-board execution, especially under unusual off-nominal circumstances

Furthermore, the study of robust optimal control and coordination of mobile robots carried out under the project will enhance C^4I capabilities of DoD for completely autonomous or command-aided remote mission execution in uncertain and dynamically evolving combat environments. The research project has pursued technology transition and develop multidisciplinary educational and training programs for academia and industry, which are vital to DoD.

On the theoretical side of the project, a rigorous methodology has been developed for automated modeling of observed behavior of autonomous agents and designing optimal mission strategies via a novel measure-theoretic optimization of probabilistic finite state behavior generators. The past decade has witnessed the development of a range of methodologies for design and control of autonomous systems, ranging from model-based to purely reactive paradigms. One of these approaches, *Probabilistic Robotics*, has led to implementations with significant autonomy and robustness. The research carried out under this project enhances the state of the art in *Probabilistic Robotics* by addressing the following key issues:

- **Probabilistic Perception:** Robots are inherently uncertain about the state of their environments. Uncertainty arises from sensor limitations, noise, and the fact that most interesting environments are, to a certain degree, unpredictable. Moreover, *a probabilistic robot knows about its own ignorance* - a key prerequisite of true autonomy.
- **Probabilistic Control:** Autonomous robots must act in the face of uncertainty. Probabilistic decision-making approaches take the robot's uncertainty into account; some consider only the robot's current uncertainty, others anticipate future uncertainty.

LIST OF PARTICIPATING SCIENTIFIC PERSONNEL

PROJECT MANAGER

Dr. David Christopher Arney
Chief, Mathematics Division
Army Research Office, Research Triangle park, NC

PRINCIPAL INVESTIGATOR

Prof. Asok Ray
Distinguished Professor of Mechanical Engineering
Pennsylvania State University, University Park, PA

KEY TECHNICAL PERSONNEL

Dr. Ishanu Chattopadhyay
Post-Doctoral Scholar, Mechanical Engineering
Pennsylvania State University, University Park, PA

LIST OF PUBLICATIONS UNDER ARO SPONSORSHIP

(a) Papers published in peer-reviewed journals

1. I. Chattopadhyay and A. Ray, “Structural transformations of probabilistic finite state machines,” ***Int. Journal of Control***, 2007, in press.
2. I. Chattopadhyay and A. Ray, "Language-Measure-Theoretic Optimal Control of Probabilistic Finite-State Systems," ***Int. Journal of Control***, Vol. 80, No. 8, August 2007, pp. 1271-1290.
3. I. Chattopadhyay and A. Ray, "Generalized Language Measure Families of Probabilistic Finite State Systems," ***Int. Journal of Control***, Vol. 80, No. 5, May 2007, pp. 789-799.

(b) Papers published in non-peer-reviewed journals or in refereed conference proceedings

1. I. Chattopadhyay and A. Ray, “Generalized Unobservability Maps in DES”, ***American Control Conference***, July 11-13, 2007, New York City, NY.

(c) Manuscripts submitted for review & In preparation.

1. G. Mallapragada, I. Chattopadhyay and A. Ray, “Autonomous Navigation of Mobile Robots Using Optimal Control of Finite State Automata," ***Int. Journal of Control***, under review .
2. G. Mallapragada, I. Chattopadhyay and A. Ray,” Automated Behavior Recognition in Mobile Robots Using Symbolic Dynamic Filtering,” ***Robotics and Autonomous Systems***, under review..
3. I. Chattopadhyay and A. Ray, “Generalized Unobservability in DES & Decidability of State Determinacy”, ***Int. Journal of Control***, under review.
4. I. Chattopadhyay, G. Mallapragada and A. Ray, “ν*: Robust Intelligent Path-planning Via Linguistic Optimization," in preparation.

Reprints of three peer reviewed journal publications

Reprints of three peer reviewed journal publications

To appear in *International Journal of Control*

Structural Transformations of Probabilistic Finite State Machines*

Ishanu Chattopadhyay‡ ixc128@psu.edu

Asok Ray‡ axr2@psu.edu

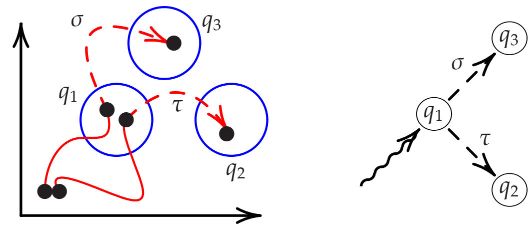
***Abstract*— Probabilistic finite state machines have recently emerged as a viable tool for modeling and analysis of complex non-linear dynamical systems. This paper rigorously establishes such models as finite encodings of probability measure spaces defined over symbol strings. The well known Nerode equivalence relation is generalized in the probabilistic setting and pertinent results on existence and uniqueness of minimal representations of probabilistic finite state machines are presented. The binary operations of probabilistic synchronous composition and projective composition are introduced, which have applications in symbolic model-based supervisory control and in symbolic pattern recognition problems. The results are elucidated with numerical examples and are validated on experimental data for statistical pattern classification in a laboratory environment.**

***Index Terms*— Formal Language Theory; Probabilistic Finite State Automata; Minimization of Probabilistic Automata; Model Order Reduction**

1. INTRODUCTION & MOTIVATION

Probabilistic finite state machines have recently emerged as a modeling paradigm for constructing causal models of complex dynamics. The general inapplicability of classical identification algorithms in complex non-linear systems has led to development of several techniques for construction of probabilistic representations of dynamical evolution from observed system behavior. The essential feature of a majority of such reported approaches is partial or complete departure from the classical continuous-domain modeling towards a formal language theoretic and hence symbolic paradigm [1][2]. The continuous range of a sufficiently long observed data set is discretized and tagged with labels to obtain a symbolic sequence [2], which is subsequently used to compute a language-theoretic finite state probabilistic predictor via recursive model update algorithms. Symbolization essentially discretizes the continuous state space and gives rise to probabilistic dynamics from the underlying deterministic process, as illustrated in Fig. 1.

Among various reported symbolic reconstruction algorithms, Causal-State Splitting Reconstruction (CSSR) [1] computes optimal representations (e.g., ϵ-machines) and is reported to yield the minimal representation consistent with accurate prediction. In contrast, the D-Markov construction [2] produces a sub-optimal model, but it has a significant computational advantage and has been shown to be better suited for online detection of small parametric anomalies in dynamic behavior of physical processes [3].



(a) (b)

Fig. 1. Emergence of probabilistic dynamics from the underlying deterministic system due to discretization as symbolic states q_1, q_2, q_3; and σ, τ are symbolic events

This paper addresses the issue of structural manipulation of such inferred probabilistic models of system dynamics. The ability to transform and manipulate the automaton structure is critical for design of supervisory control algorithms for symbolic models and real-time pattern recognition from symbol sequences. Specific issues are delineated in the sequel.

A. Applications of Symbolic Model-based Control

The natural setting for developing control algorithms for symbolic models is that of probabilistic languages. The notion of probabilistic languages in the context of studying qualitative stochastic behavior of discrete-event systems first appeared in [4] and [5], where the concept of p-languages ('p' implying probabilistic) is introduced and an algebra is developed to model probabilistic languages based on concurrency. A multitude of control algorithms for p-language-theoretic models have been reported. Earlier approaches [6][7] attempt a direct generalization of Ramadge and Wonham's Supervisory Control Theory [8] for deterministic languages and proves to be somewhat cumbersome in practice. A significantly simpler approach is suggested in [9][10][11], where supervisory control laws are synthesized by elementwise maximization of a language measure vector [12][9] to ensure that the generated event strings cause the supervised plant to visit the "good" states while attempting to avoid the "bad" states optimally in a probabilistic sense. The notion of "good" and "bad" is induced by specifying scalar weights on the

*This work has been supported in part by the U.S. Army Research Office under Grant Nos. W911NF-06-1-0469 and W911NF-07-1-0376.

‡The Pennsylvania State University, University Park, PA 16802

model states, with relatively more negative weights indicating less desirable states. Unlike the previous approaches, the measure-theoretic approach does not require a "specification automaton"; however, a specification weight is assigned to each state of the finite state machine. (Note: These states are different from the states obtained via symbolic reconstruction of observed physical data.)

Figure 2 illustrates the underlying concept. The symbolic model shown on the left which has three states q_1, q_2, q_3, while the control objective is specified by weights +1 and −1 on states q_A, q_B of the two-state automaton on the right.

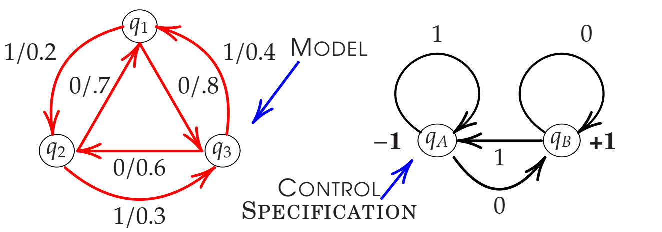


Fig. 2. Symbolic Model and Control Specification

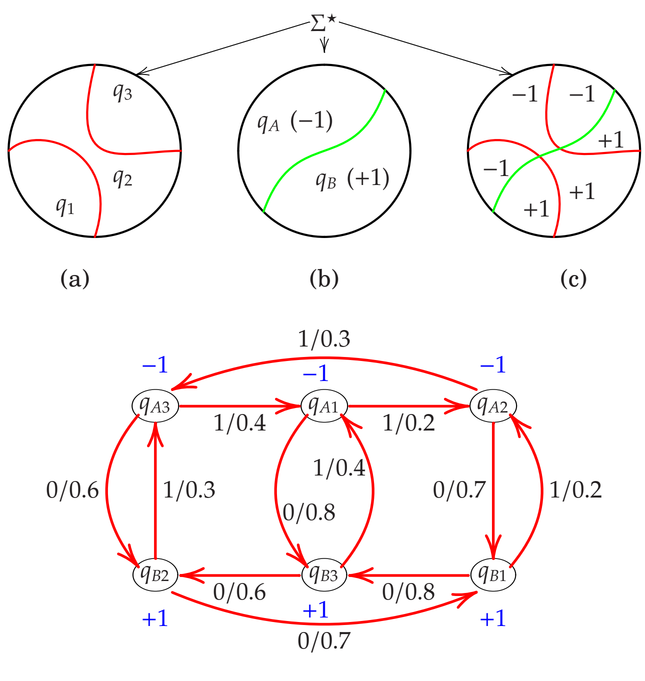


Fig. 3. Imposing control specification by probabilistic synchronous composition of automata

Recalling that every finite state automaton induces a right invariant partition on the set of all possible finite length strings, the above situation is illustrated by Figs. 3(a) and 3(b). The operation of probabilistic synchronous composition, defined later in this paper, resolves the problem by considering the product partition in Fig. 3(c). Then, the given model is transformed into the one shown in Fig. 3(d), on which the optimization algorithm reported in [11] can be directly applied to yield the optimal supervision policy.

B. Applications of Symbolic Pattern Recognition

As mentioned earlier, the symbolic reconstruction algorithms [1][2] generate probabilistic finite state models from observed time series. However, in a pattern classification problem, one may be only interested in a given class of possible future evolutions. For example, as illustrated in Fig. 4, while the systems $G_1, G_2, \cdots, G_k$ yield different symbolic models $A_1, A_2, \cdots, A_k$, we may be only interested in matching a given template, i.e., knowing how similar the systems are as far as strings with even number of 0s is concerned (Note: q_A = {strings with even number of 0s}).

The operation of projective composition, defined in this paper, allows transformation of each model A_i to the structure of the template while preserving the distribution over the strings of interest, and is of critical importance in symbolic pattern classification problems. As shown in the sequel, the model order of the machines A_i is not particularly important; hence projective composition accomplishes model order reduction within a quantifiable error.

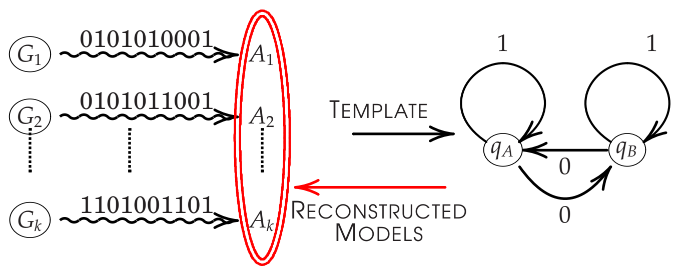


Fig. 4. Symbolic template matching problem

C. Organization of the Paper

The paper is organized in seven sections including the present one. Section 2 presents preliminary concepts and pertinent results that are necessary for subsequent development. Section 3 introduces the concept of probabilistic finite state automata as finite encodings of probability measure spaces. The concept of Nerode equivalence is generalized to probabilistic automata and the key results on existence and uniqueness of minimal representations are established. Section 4 presents metrics on the space of probability measures on symbolic strings which is shown to induce pseudometrics on the space of probabilistic finite state automata. Along this line, the concept of probabilistic synchronous composition is introduced and the results are elucidated with a simple example. Section 5 defines projective composition and invariance of projected distributions is established. A numerical example is provided for clarity of exposition. Section 6 demonstrates applicability of the developed method

to a pattern classification problem on experimental data. The paper is summarized and concluded in Section 7 with recommendations for future research.

2. Preliminary Notions

A deterministic finite state automaton (*DFSA*) is defined [13] as a quintuple $G_i = (Q, \Sigma, \delta, q_i, Q_m)$, where Q is the finite set of states, and $q_i \in Q$ is the initial state; Σ is the (finite) alphabet of events. The Kleene closure of Σ, denoted as $\Sigma^\star$, is the set of all finite-length strings of events including the empty string ε; the set of all finite-length strings of events excluding the empty string ε is denoted as Σ^+ and the set of all *strictly* infinite-length strings of events is denoted as Σ^ω. A subset of Σ^ω is called an ω-language on the alphabet Σ and a subset of $\Sigma^\star$ is called a $\star$-language. If the meaning is clear from context, we refer to a set of strings simply as a language. The function $\delta : Q \times \Sigma \to Q$ represents the state transition map and $\delta^\star : Q \times \Sigma^\star \to Q$ is the reflexive and transitive closure [13] of δ and $Q_m \subseteq Q$ is the set of marked (i.e. accepting) states. For given functions f and g, we denote the composition as $f \circ g$.

Definition 2.1: *The classical Nerode equivalence $\mathcal{N}$ [13] on $\Sigma^\star$ with respect to a given language L is defined as:*

$$\forall x, y \in \Sigma^\star, \left(x\mathcal{N}y \Leftrightarrow \left(\forall u \in \Sigma^* \left(\; xu \in L \right) \iff \left(yu \in L \right) \right) \right) \quad (1)$$

A language $L \subseteq \Sigma^\star$ is regular if and only if the corresponding Nerode equivalence is of finite index [13].

Probabilistic Finite State automata (PFSA) considered in this paper are built upon Deterministic Finite State Automata (DFSA) with a specified event generating function. The formal definition is stated next.

Definition 2.2: (PFSA) *A probabilistic finite state automata (PFSA) is a quintuple $P_i \triangleq (Q, \Sigma, \delta, q_i, \tilde{\pi})$ where the quadruple (Q, Σ, δ, q_i) is a DFSA with unspecified marked states and the mapping $\tilde{\pi} : Q \times \Sigma \to [0, 1]$ satisfies the following condition:*

$$\forall q_j \in Q, \sum_{\sigma \in \Sigma} \tilde{\pi}(q_j, \sigma) = 1 \quad (2)$$

In the sequel, $\tilde{\pi}$ is denoted as the event generating function. For a PFSA P_i, cardinality of the set of states is denoted as $\textsc{NumStates}(P_i)$.

Definition 2.3: *For every PFSA $P_i = (Q, \Sigma, \delta, q_i, \tilde{\pi})$, there is an associated stochastic matrix $\Pi \in \mathbb{R}^{\textsc{NumStates}(P_i) \times \textsc{NumStates}(P_i)}$, called the state transition probability matrix, which is defined as follows:*

$$\Pi_{jk} = \sum_{\sigma : \delta(q_j, \sigma) = q_k} \tilde{\pi}(q_j, \sigma) \quad (3)$$

We further note that for every stochastic matrix Π, there exists at least one row-vector $\wp$ such that

$$\wp \Pi = \wp, \text{ where } \forall j \; \wp_j \geqq 0 \text{ and } \sum_{j=1}^{\textsc{NumStates}(P_i)} \wp_j = 1 \quad (4)$$

where $\wp$ is a stable long term distribution over the PFSA states. If Π is irreducible, then $\wp$ is unique. Otherwise, there may exist more than one possible solution to Eq. (4), one for each eigenvector corresponding to unity eigenvalue. However, if the initial state is specified (as it is in this paper), then $\wp$ is always unique. Several efficient algorithms have been reported in the literature [14][15][16] for computation of $\wp$.

Key definitions and results from Measure theory that are used here are recalled.

Definition 2.4: (σ-Algebra) *A collection $\mathfrak{M}$ of subsets of a non-empty set X is said to be a σ-algebra [17] in X if $\mathfrak{M}$ has the following properties:*

1) $X \in \mathfrak{M}$
2) *If $A \in \mathfrak{M}$, then $A^c \in \mathfrak{M}$ where A^c is the complement of A relative to X, i.e., $A^c = X \setminus A$*
3) *If $A = \bigcup_{n=1}^{\infty} A_n$ and if $A_n \in \mathfrak{M}$ for $n \in \mathbb{N}$, then $A \in \mathfrak{M}$.*

Theorem 2.1: *If $\mathscr{F}$ is any collection of subsets of X, there exists a smallest σ-algebra $\mathfrak{M}^\star$ in X such that $\mathscr{F} \subseteq \mathfrak{M}^\star$.*

Proof: See Theorem 1.10 in [17]. ❒

Definition 2.5: (Measure) *A finite (non-negative) measure is a countably additive function μ, defined on a σ-algebra $\mathfrak{M}$, whose range is $[0, K]$ for some $K \in \mathbb{R}$. Countable additivity means that if $\{A_i\}$ is a disjoint countable collection of members of $\mathfrak{M}$, then*

$$\mu\left(\bigcup_{i=1}^{\infty} A_i \right) = \sum_{i=1}^{\infty} \mu(A_i) \quad (5)$$

Theorem 2.2: *If μ is a (non-negative) measure on a σ-algebra $\mathfrak{M}$, then*

1) $\mu(\varnothing) = 0$
2) ***(Monotonicity)*** *$A \subseteq B \Longrightarrow \mu(A) \leq \mu(B)$ if $A, B \in \mathfrak{M}$.*

Proof: See Theorem 1.19 in [17]. ❒

Definition 2.6: *A probability measure on a non-empty set with a specified σ-algebra $\mathfrak{M}$ is a finite non-negative measure on $\mathfrak{M}$. Although not required by the theory, a probability measure is defined to have the unit interval $[0, 1]$ as its range.*

Definition 2.7: *A probability measure space is a triple $(X, \mathfrak{M}, \mathfrak{p})$ where X is the underlying set, $\mathfrak{M}$ is the σ-algebra in X and $\mathfrak{p}$ is a finite non-negative measure on $\mathfrak{M}$.*

3. Properties of Probabilistic Finite State Automata

For any $\tau \in \Sigma^\star$, the language $\tau\Sigma^\omega$ has an important physical interpretation pertaining to systems modeled as probabilistic language generators (See Fig. 5). A string $\tau \in \Sigma^\star$ can be interpreted as a symbol sequence that has been already generated, and any string in Σ^ω qualifies as a possible

future evolution. Thus, the language $\tau\Sigma^\omega$ is conceptually associated with the current dynamical state of the modeled system.

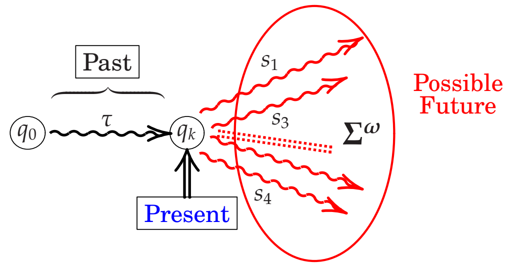


Fig. 5. Interpretation of the language $\tau\Sigma^\omega$ pertaining to dynamical evolution of a language generator

Definition 3.1: *Given an alphabet Σ, the set $\mathfrak{B}_\Sigma \triangleq 2^{\Sigma^\star}\Sigma^\omega$ is defined to be the σ-algebra generated by the set $\left\{L : L = \tau\Sigma^\omega \text{ where } \tau \in \Sigma^\star\right\}$, i.e., the smallest σ-algebra on the set Σ^ω, which contains the set $\left\{L : L = \tau\Sigma^\omega \text{ where } \tau \in \Sigma^\star\right\}$.*

Remark 3.1: *Cardinality of $\mathfrak{B}_\Sigma$ is $\aleph_1$ because both $2^{\Sigma^\star}$ and Σ^ω have cardinality $\aleph_1$.*

The following relations in the probability measure space $(\Sigma^\omega, \mathfrak{B}_\Sigma, \mathfrak{p})$ are consequences of Definition 3.1.

- $\mathfrak{p}(\Sigma^\omega) = \mathfrak{p}(\varepsilon\Sigma^\omega) = 1$
- $\forall x, u \in \Sigma^\star, xu\Sigma^\omega \subseteq x\Sigma^\omega$ and hence $\mathfrak{p}(xu\Sigma^\omega) \leqq \mathfrak{p}(\Sigma^\omega)$

Notation 3.1: *For brevity, the probability $\mathfrak{p}(\tau\Sigma^\omega)$ is denoted as $\mathfrak{p}(\tau)$ $\forall\tau \in \Sigma^\star$ in the sequel.*

Next the notion of probabilistic Nerode equivalence $\mathcal{N}_\mathfrak{p}$ is introduced on $\Sigma^\star$ for representing the measure space $(\Sigma^\omega, \mathfrak{B}_\Sigma, \mathfrak{p})$ in the form of a PFSA. In this context, the following logical formulae are introduced.

Definition 3.2: *For $x, y \in \Sigma^\star$,*

$$\mathbb{U}_1(x, y) \triangleq \Big(\mathfrak{p}(x) = 0 \bigwedge \mathfrak{p}(y) = 0\Big) \tag{6a}$$

$$\mathbb{U}_2(x, y) \triangleq \Big(\mathfrak{p}(x) \neq 0 \bigwedge \mathfrak{p}(y) \neq 0\Big) \bigwedge \left(\forall u \in \Sigma^\star \left(\frac{\mathfrak{p}(xu)}{\mathfrak{p}(x)} = \frac{\mathfrak{p}(yu)}{\mathfrak{p}(y)}\right)\right) \tag{6b}$$

Theorem 3.1: (Probabilistic Nerode Equivalence) *Given an alphabet Σ, every measure space $(\Sigma^\omega, \mathfrak{B}_\Sigma, \mathfrak{p})$ induces a right-invariant equivalence relation $\mathcal{N}_\mathfrak{p}$ on $\Sigma^\star$ defined as:*

$$\forall x, y \in \Sigma^\star, \left(x\mathcal{N}_\mathfrak{p}y \Longleftrightarrow \mathbb{U}_1(x, y) \bigvee \mathbb{U}_2(x, y)\right) \tag{7}$$

Proof: Reflexivity and symmetry properties of the relation $\mathcal{N}_\mathfrak{p}$ follow from Definition 3.2. Let $x, y, z \in \Sigma^\star$ be distinct and arbitrary strings such that $x\mathcal{N}_\mathfrak{p}y$ and $y\mathcal{N}_\mathfrak{p}z$. Then, transitivity property of $\mathcal{N}_\mathfrak{p}$ follows from Eq. (7) and Definition 3.2. Hence, $\mathcal{N}_\mathfrak{p}$ is an equivalence relation.

To establish right-invariance [13] of $\mathcal{N}_\mathfrak{p}$, it suffices to show that

$$\forall x, y \in \Sigma^\star, \left(x\mathcal{N}_\mathfrak{p}y \Longrightarrow \forall u \in \Sigma^\star, \left(xu\mathcal{N}_\mathfrak{p}yu\right)\right) \tag{8}$$

Let x, y, u be arbitrary strings in $\Sigma^\star$ such that $x\mathcal{N}_\mathfrak{p}y$. If $\mathfrak{p}(x) = 0$, $\mathfrak{p}(y) = 0$ from Eq. (7). Then, it follows from the monotonicity property of the measure (See Theorem 2.2) that $\mathfrak{p}(xu) = 0$, which implies the truth of $\mathbb{U}_1(xu, yu)$ and hence the truth of $xu\mathcal{N}_\mathfrak{p}yu$. If $\mathfrak{p}(x) \neq 0$, then $\left(x\mathcal{N}_\mathfrak{p}y\right) \bigwedge \left(\mathfrak{p}(x) \neq 0\right)$ implies $\mathfrak{p}(y) \neq 0$. Hence,

$$\frac{\mathfrak{p}(xu\tau)}{\mathfrak{p}(xu)} = \frac{\mathfrak{p}(xu\tau)}{\mathfrak{p}(x)} \times \frac{\mathfrak{p}(x)}{\mathfrak{p}(xu)} \tag{9}$$

If $\mathfrak{p}(x) = \mathfrak{p}(y)$, then $x\mathcal{N}_\mathfrak{p}y$ implies $\mathfrak{p}(xu) = \mathfrak{p}(yu)$ and also $\forall\tau \in \Sigma^\star\big(\mathfrak{p}(xu\tau) = \mathfrak{p}(yu\tau)\big)$. Similarly, if $\mathfrak{p}(x) \neq \mathfrak{p}(y)$, then $x\mathcal{N}_\mathfrak{p}y$ implies $\mathfrak{p}(xu) \neq \mathfrak{p}(yu)$ and also $\forall\tau \in \Sigma^\star\big(\mathfrak{p}(xu\tau) \neq \mathfrak{p}(yu\tau)\big)$. Hence, $\forall\tau \in \Sigma^\star\Big(\big(\mathfrak{p}(xu) = \mathfrak{p}(yu)\big) \iff \big(\mathfrak{p}(xu\tau) = \mathfrak{p}(yu\tau)\big)\Big)$. ❒

Definition 3.3: (Perfect Encoding) *Given an alphabet Σ, PFSA $P_i = (Q, \Sigma, \delta, q_i, \tilde{\pi})$ is defined to be a perfect encoding of the measure space $(\Sigma^\omega, \mathfrak{B}_\Sigma, \mathfrak{p})$ if $\forall\tau \in \Sigma^+$ and $\tau = \sigma_1\sigma_2\cdots\sigma_r$,*

$$\mathfrak{p}(\tau) = \tilde{\pi}(q_i, \sigma_1)\prod_{k=1}^{r-1}\tilde{\pi}(\delta^\star(q_i, \sigma_1\cdots\sigma_k), \sigma_{k+1}) \tag{10}$$

Remark 3.2: *The implications of Definition 3.3 are as follows: The encoding introduced is perfect in the sense that the measure $\mathfrak{p}$ can be reconstructed without error from the specification of P_i.*

Theorem 3.2: *A PFSA is a perfect encoding if and only if the corresponding probabilistic Nerode equivalence $\mathcal{N}_\mathfrak{p}$ is of finite index.*

Proof: *(Left to Right:)* Let Q be the finite set of equivalence classes of the relation $\mathcal{N}_\mathfrak{p}$ of the PFSA $P_i = (Q, \Sigma, \delta, q_i, \tilde{\pi})$ that is constructed as follows:

1) Since $\mathcal{N}_\mathfrak{p}$ is an equivalence relation on $\Sigma^\star$, there exists a unique $q_i \in Q$ such that $\varepsilon \in q_i$. The initial state of P_i is set to q_i.
2) If $x \in q_j$ and $x\sigma \in q_k$, then $\delta(q_j, \sigma) = q_k$
3) $\tilde{\pi}(q_j, \sigma) = \frac{\mathfrak{p}(x\sigma)}{\mathfrak{p}(x)}$ where $x \in q_j$.

First we verify that the steps 2 and 3 are consistent in the sense that δ and $\tilde{\pi}$ are well-defined.

Probabilistic Nerode equivalence (See Theorem 3.1) implies that if $x, y \in \Sigma^\star$, then $\Big(\big(x \in q_j\big) \wedge (x\sigma \in q_k) \wedge \big(y \in q_j\big)\Big) \Longrightarrow \big(y\sigma \in q_k\big)$. Therefore, the constructed δ is well-defined. Similarly, since $(x, y \in q_j) \Longrightarrow \big(\mathfrak{p}(x) = \mathfrak{p}(y)\big) \bigwedge \big(\mathfrak{p}(x\sigma) = \mathfrak{p}(y\sigma)\big)$, the constructed $\tilde{\pi}$ is also well-defined. Therefore, the steps 2 and 3 are consistent. For $\tau = \sigma_1\sigma_2\cdots\sigma_r \in \Sigma^+$, it follows that

$$\mathfrak{p}(\tau) = p(\sigma_1)\prod_{r=2}^{R}\frac{\mathfrak{p}(\sigma_1\cdots\sigma_r)}{\mathfrak{p}(\sigma_1\cdots\sigma_{r-1})}$$
$$= \tilde{\pi}(q_i, \sigma_1)\prod_{r=1}^{R-1}\tilde{\pi}(\delta^\star(q_i, \sigma_1\cdots\sigma_r), \sigma_{r+1})$$

Hence, the criterion for perfect encoding (See Definition 3.3) is satisfied.

(Right to Left:) Let the PFSA $P_i = (Q, \Sigma, \delta, q_i, \tilde{\pi})$ be a perfect encoding; and let the probabilistic Nerode equivalence $\mathcal{N}_\mathfrak{p}$ be of infinite index. Then, there exists a set of strings $\mathscr{H} \subseteq \Sigma^\star$,

having the same cardinality as $\Sigma^\star$, such that each element of $\mathscr{H}$ belongs to a distinct $\mathcal{N}_\mathfrak{p}$-equivalence class. That is, $\forall h_j, h_k \in \mathscr{H}$ such that $j \neq k$, we have $h_j \not\mathcal{N}_\mathfrak{p}\, h_k$. Since $\mathfrak{p}(h_j) = \mathfrak{p}(h_k) = 0$ implies $h_j \mathcal{N}_\mathfrak{p} h_k$, there can exist at most one element $h_0 \in \mathscr{H}$ such that $\mathfrak{p}(h_0) = 0$. That is, $\mathfrak{p}(h_j) \neq 0 \ \forall h_j \in \mathscr{H} - \{h_0\}$.

For the PFSA $P_i = (Q, \Sigma, \delta, q_i, \tilde{\pi})$, where Q is the finite set of states, there exists $q_\ell \in Q$ and $h_j, h_k \in \mathscr{H}$ such that $\delta^\star(q_i, h_j) = \delta^\star(q_i, h_k) = q_\ell$. Let $\tau \in \Sigma^+$ and $\tau = \sigma_1\sigma_2\cdots\sigma_r$. Since P_i ia a perfect encoding, it follows from Definition 3.3 that

$$\mathfrak{p}(h_j\tau) = \mathfrak{p}(h_j)\tilde{\pi}(q_\ell, \sigma_1)\prod_{m=1}^{r-1}\tilde{\pi}(\delta^\star(q_\ell, \sigma_1\cdots\sigma_m), \sigma_{m+1})$$

$$\mathfrak{p}(h_k\tau) = \mathfrak{p}(h_k)\tilde{\pi}(q_\ell, \sigma_1)\prod_{m=1}^{r-1}\tilde{\pi}(\delta^\star(q_\ell, \sigma_1\cdots\sigma_m), \sigma_{m+1})$$

Now, it follows that

$$\Big(\mathfrak{p}(h_j) \neq 0 \wedge \mathfrak{p}(h_k) \neq 0\Big) \bigwedge \left(\frac{\mathfrak{p}(h_j\tau)}{\mathfrak{p}(h_j)} = \frac{\mathfrak{p}(h_k\tau))}{\mathfrak{p}(h_k)}\right)$$
$$\Longrightarrow \mathbb{U}_2(h_j, h_k) \Longrightarrow h_j \mathcal{N}_\mathfrak{p} h_k$$

which contradicts the initial assertion that $h_j \not\mathcal{N}_\mathfrak{p}\, h_k \ \forall h_j, h_k \in \mathscr{H}$. This completes the proof. ❐

The construction in the first part of Theorem 3.2 is stated in the form of Algorithm 1.

Algorithm 1: Construction of PFSA from the probability measure space $(\Sigma^\omega, \mathfrak{B}_\Sigma, \mathfrak{p})$

input : $(\Sigma^\omega, \mathfrak{B}_\Sigma, \mathfrak{p})$ such that $\mathcal{N}_\mathfrak{p}$ is of finite index
output: P_i
begin
  Let $Q = \left\{q_j : j \in \mathcal{J} \subsetneq \mathbb{N}\right\}$ be the set of equivalence classes of the relation $\mathcal{N}_\mathfrak{p}$;
  Set the initial state of P_i as q_i such that ε belongs to the equivalence class q_i; If $x \in q_j$ and $x\sigma \in q_k$, then set $\delta(q_j, \sigma) = q_k$;
  $\tilde{\pi}(q_j, \sigma) = \frac{\mathfrak{p}(x\sigma)}{\mathfrak{p}(x)}$ where $x \in q_j$;
end

Corollary 3.1: (to Theorem 3.2) *A PFSA $P_i = (Q, \Sigma, \delta, q_i, \tilde{\pi})$ induces a probability measure $\mathfrak{p}$ on the σ-algebra $\mathfrak{B}_\Sigma$ and the corresponding probabilistic Nerode equivalence is of finite index.*

Proof: Let a probability measure $\mathfrak{p}$ be constructed on the σ-algebra $\mathfrak{B}_\Sigma$ as follows:

$$\forall \tau \in \Sigma^+,\ \left(\mathfrak{p}(\tau) = \tilde{\pi}(q_i, \sigma_1)\prod_{k=1}^{r-1}\tilde{\pi}(\delta^\star(q_i, \sigma_1\cdots\sigma_k), \sigma_{k+1})\right)$$

It follows from Definition 3.3 that P_i perfectly encodes the measure $\mathfrak{p}$ and Theorem 3.2 implies that the corresponding $\mathcal{N}_\mathfrak{p}$ is of finite index. ❐

On account of Corollary 3.1, we can map any given PFSA to a measure space $(\Sigma^\omega, \mathfrak{B}_\Sigma, \mathfrak{p})$.

Definition 3.4: *Let $\mathscr{P}$ be the space of all probability measures on $\mathfrak{B}_\Sigma$ and $\mathscr{A}$ be the space of all possible PFSA $P_i = (Q, \Sigma, \delta, q_i, \tilde{\pi})$.*

- *The map $\mathbb{H} : \mathscr{A} \to \mathscr{P}$ is defined as $\mathbb{H}(P_i) = \mathfrak{p}$ such that*

$$\forall \tau \in \Sigma^+,\ \left(\mathfrak{p}(\tau) = \tilde{\pi}(q_i, \sigma_1)\prod_{k=1}^{r-1}\tilde{\pi}(\delta^\star(q_i, \sigma_1\cdots\sigma_k), \sigma_{k+1})\right)$$
$$\text{where } \tau = \sigma_1\sigma_2\cdots\sigma_r \tag{11}$$

- *The map $\mathbb{H}_{-1} : \mathscr{P} \to \mathscr{A}$ is defined as:*

$$\mathbb{H}_{-1}(\mathfrak{p}) = \begin{cases} P_i \text{ given by Algo. } 1 & \text{if } \mathcal{N}_\mathfrak{p} \text{ is of finite index} \\ \text{Undefined} & \text{otherwise} \end{cases} \tag{12}$$

Lemma 3.1: *P_i is a perfect encoding for $\mathbb{H}(P_i)$.*

Proof: The proof follows from Definition 3.3 and Definition 3.4. ❐

Next we show that, similar to classical finite state machines, an arbitrary PFSA can be uniquely minimized. However, the sense in which the minimization is achieved is somewhat different. To this end, we introduce the notion of reachable states in a PFSA and define isomorphism of two PFSA.

Definition 3.5: (Reachable States) *Given a PFSA $P_i = (Q, \Sigma, \delta, q_i, \tilde{\pi})$, the set of reachable states $\mathsf{RCH}(P_i) \subsetneqq Q$ is defined as:*

$$\tilde{q} \in \mathsf{RCH}(P_i) \Longrightarrow \exists \tau = \sigma_1\cdots\sigma_R \in \Sigma^\star \textit{ such that}$$
$$\Big(\delta^\star(q_i, \tau) = \tilde{q}\Big) \bigwedge \left(\tilde{\pi}(q_i, \sigma_1)\prod_{r=1}^{R-1}\tilde{\pi}(\delta^\star(q_i, \sigma_1\cdots\sigma_r), \sigma_{r+1}) > 0\right)$$

Remark 3.3: *The strict positivity condition in Definition 3.5 ensures that every state in the set of reachable states can actually be attained with a strictly non-zero probability. In other words, for every state $q_j \in \mathsf{RCH}(P_i)$, there exists at least one string ω, initiating from q_i and eventually terminating on state q_j, such that the generation probability of ω is strictly positive.*

Definition 3.6: (Isomorphism:) *Two PFSA $P_i = (Q, \Sigma, \delta, q_i, \tilde{\pi})$ and $P'_{i'} = (Q', \Sigma, \delta', q'_{i'}, \tilde{\pi}')$ are defined to be isomorphic if there exists a bijective map $\eta : \mathsf{RCH}(P_i) \longrightarrow \mathsf{RCH}(P'_{i'})$ such that*

$$\tilde{\pi}(q_j, \sigma) \neq 0 \Rightarrow \Big(\tilde{\pi}'(\eta(q_j), \sigma) = \tilde{\pi}(q_j, \sigma)\Big) \bigwedge$$
$$\Big(\delta'(\eta(q_j), \sigma_k) = \eta(\delta(q_j, \sigma_k))\Big)$$

Remark 3.4: *The notion of isomorphism stated in Definition 3.6 generalizes graph isomorphism to PFSA by considering only the states that can be reached with non-zero probability and transitions that have a non-zero probability of occurrence.*

Theorem 3.3: (Minimization of PFSA:) *For a PFSA $P_i = (Q, \Sigma, \delta, q_i, \tilde{\pi})$, $\mathbb{H}_{-1} \circ \mathbb{H}(P_i)$ is the unique minimal realization of P_i in the sense that the following conditions are satisfied:*

1) *The PFSA* $\mathbb{H}_{-1} \circ \mathbb{H}(P_i)$ *perfectly encodes the probability measure* $\mathbb{H}(P_i)$.
2) *For a PFSA* $P'_{i'}$ *that perfectly encodes* $\mathbb{H}(P_i)$, *the inequality* $\textsc{Card}(\textsc{Rch}(\mathbb{H}_{-1} \circ \mathbb{H}(P_i))) \leqq \textsc{Card}(\textsc{Rch}(P'_{i'}))$ *holds.*
3) *The equality,* $\textsc{Card}(\textsc{Rch}(\mathbb{H}_{-1} \circ \mathbb{H}(P_i))) = \textsc{Card}(\textsc{Rch}(P'_{i'}))$, *implies isomorphism of* P_i *and* $P'_{i'}$ *in the sense of Definition 3.6.*

Proof:

1) The proof follows from the construction in Theorem 3.2.
2) Let $P'_{i'} = (Q', \Sigma, \delta', q_{i'}, \tilde{\pi}')$ be an arbitrary PFSA that perfectly encodes the probability measure $\mathbb{H}(P_i)$. Let us construct a PFSA $P^\dagger_{i'} = (Q' \cup \{q_d\}, \Sigma, \delta^\dagger, q_{i'}, \tilde{\pi}^\dagger)$, where q_d is a new state not in Q', as follows:

$$\forall q'_j \in Q', \sigma \in \Sigma,$$

$$\delta^\dagger(q'_j, \sigma_k) = \begin{cases} q_d & \text{if } \tilde{\pi}'(q'_j, \sigma_k) = 0 \\ \delta'(q'_j, \sigma_k) & \text{otherwise} \end{cases} \tag{13a}$$

$$\forall \sigma \in \Sigma,\ \delta^\dagger(q_d, \sigma_k) = q_d \tag{13b}$$

$$\forall q'_j \in Q',\ \forall \sigma \in \Sigma,\ \tilde{\pi}^\dagger(q'_j, \sigma_k) = \tilde{\pi}'(q'_j, \sigma_k) \tag{13c}$$

It is seen that $P^\dagger_{i'}$ perfectly encodes $\mathbb{H}(P_i)$ as well, which follows from Definition 3.3 and Eq. (13c). It is claimed that

$$\textsc{Card}(\textsc{Rch}(P^\dagger_{i'})) = \textsc{Card}(\textsc{Rch}(P'_{i'})) \tag{14}$$

based on the following rationale.

Let $q'_j \in \textsc{Rch}(P'_{i'})$. Following Definition 3.5, there exists a string $\tau \in \Sigma^\star$ such that $\delta'^\star(q_{i'}, x) = q'_j$ and $\tilde{\pi}'(q_{i'}, \sigma_1) \prod_{r=1}^{R-1} \tilde{\pi}'(\delta'^\star(q_{i'}, \sigma_1 \cdots \sigma_r), \sigma_{r+1}) > 0$. It follows from Eq. (13c) that $\tilde{\pi}^\dagger(q_{i'}, \sigma_1) \prod_{r=1}^{R-1} \tilde{\pi}^\dagger(\delta^{\dagger\star}(q_{i'}, \sigma_1 \cdots \sigma_r), \sigma_{r+1}) > 0$ and hence we conclude using Eq. (13a) that $\delta^{\dagger\star}(q_{i'}, x) = q_j \neq q_d$ which then implies that $q'_j \in \textsc{Rch}(P^\dagger_{i'})$. Hence we have $\textsc{Card}(\textsc{Rch}(P'_{i'})) \leqq \textsc{Card}(\textsc{Rch}(P^\dagger_{i'}))$.By a similar argument, we have $\textsc{Card}(\textsc{Rch}(P^\dagger_{i'})) \leqq \textsc{Card}(\textsc{Rch}(P'_{i'}))$ and hence $\textsc{Card}(\textsc{Rch}(P^\dagger_{i'})) = \textsc{Card}(\textsc{Rch}(P'_{i'}))$.

Next, we claim

$$\forall x, y \in \Sigma^\star \Big(\big(\delta^\dagger(q_{i'}, x) = \delta^\dagger(q_{i'}, y)\big) \Longrightarrow x \mathcal{N}_{\mathbb{H}(P_i)} y \Big) \tag{15}$$

based on the following rationale.

Let $x, y \in \Sigma^\star$ s.t. $\big(\delta^\dagger(q_{i'}, x) = \delta^\dagger(q_{i'}, y)\big)$. It follows from Eqs. (13a), (13b) and (13c) that

$$\begin{cases} \big(\mathbb{H}(P_i)(x) = 0 \wedge \mathbb{H}(P_i)(y) = 0\big) \text{ if } \delta^\dagger(q_{i'}, x) = q_d \\ \big(\mathbb{H}(P_i)(x) \neq 0 \wedge \mathbb{H}(P_i)(y) \neq 0\big) \text{ otherwise} \end{cases} \tag{16}$$

Now, if $\big(\mathbb{H}(P_i)(x) = 0 \wedge \mathbb{H}(P_i)(y) = 0\big)$, then it follows from Eq. (6b) that $x \mathcal{N}_{\mathbb{H}(P_i)} y$. On the other hand, if $\big(\mathbb{H}(P_i)(x) \neq 0 \wedge \mathbb{H}(P_i)(y) \neq 0\big)$, then Eq. (6a) yields:

$$\forall u = \sigma_1 \cdots \sigma_R \in \Sigma^\star,\ \frac{\mathbb{H}(P_i)(xu)}{\mathbb{H}(P_i)(x)} = \frac{\mathbb{H}(P_i)(yu)}{\mathbb{H}(P_i)(y)}$$

$$= \tilde{\pi}^\dagger(q_{i'}, \sigma_1) \prod_{r=1}^{R-1} \tilde{\pi}^\dagger(\delta^\dagger(q_{i'}, \sigma_1 \cdots \sigma_r), \sigma_{r+1}) \Rightarrow x \mathcal{N}_{\mathbb{H}(P_i)} y$$

We define a map $\zeta : \textsc{Rch}(\mathbb{H}_{-1} \circ \mathbb{H}(P_i)) \to \textsc{Rch}(P^\dagger_{i'})$ as follows: Let $q^\# \in \textsc{Rch}(\mathbb{H}_{-1} \circ \mathbb{H}(P_i))$ and let $\mathscr{E}(q^\#)$ be the equivalence class of the relation $\mathcal{N}_{\mathbb{H}(P_i)}$ represented by $q^\#$. Let $x = \sigma_1 \cdots \sigma_R \in \mathscr{E}(q^\#)$.

$$\begin{aligned} & \mathbb{H}(P_i)(x) > 0 \text{ (See Definition 3.5)} \\ \Longrightarrow\ & \tilde{\pi}^\dagger(q_{i'}, \sigma_1) \prod_{r=1}^{R-1} \tilde{\pi}^\dagger(\delta^{\dagger\star}(q_{i'}, \sigma_1 \cdots \sigma_r), \sigma_{r+1}) > 0 \\ & \qquad \text{(Since } \mathrm{P}^\dagger_{\mathrm{i}'} \text{ perfectly encodes } \mathbb{H}(\mathrm{P_i})) \\ \Longrightarrow\ & \delta^{\dagger\star}(q_{i'}, x) \in \textsc{Rch}(P^\dagger_{i'}) \end{aligned}$$

Let $\zeta(q^\#) = \delta^{\dagger\star}(q_{i'}, x)$. Note that $\zeta(q^\#)$ depends on the choice of x. Let $q^\#_1, q^\#_2 \in \textsc{Rch}(\mathbb{H}_{-1} \circ \mathbb{H}(P_i))$ such that $\zeta(q^\#_1) = \zeta(q^\#_2)$. If x_1, x_2 are the corresponding strings chosen to define $\zeta(q^\#_1), \zeta(q^\#_2)$, we have $\delta^{\dagger\star}(q_{i'}, x_1) = \delta^{\dagger\star}(q_{i'}, x_2)$ which implies $x_1 \mathcal{N}_{\mathbb{H}(P_i)} x_2$, i.e., $q^\#_1 = q^\#_2$. Hence we conclude ζ is injective which, in turn, implies

$$\textsc{Card}(\textsc{Rch}(\mathbb{H}_{-1} \circ \mathbb{H}(P_i))) \leqq \textsc{Card}(\textsc{Rch}(P^\dagger_{i'})) \tag{17}$$

Finally, from Eqs. (14) and (17), it follows that

$$\textsc{Card}(\textsc{Rch}(\mathbb{H}_{-1} \circ \mathbb{H}(P_i))) \leqq \textsc{Card}(\textsc{Rch}(P'_{i'})) \tag{18}$$

3) Let $P'_{i'} = (Q', \Sigma, \delta', q_{i'}, \tilde{\pi}')$ be an arbitrary PFSA that perfectly encodes $\mathbb{H}(P_i)$ such that

$$\textsc{Card}(\textsc{Rch}(\mathbb{H}_{-1} \circ \mathbb{H}(P_i))) = \textsc{Card}(\textsc{Rch}(P'_{i'})) \tag{\textbf{C1}}$$

Let the PFSA $\mathbb{H}_{-1} \circ \mathbb{H}(P_i)$ be denoted as $(Q^\#, \Sigma, \delta^\#, q^\#_{i^\#}, \tilde{\pi}^\#)$. Let $\mathscr{E}(q^\#_j)$ denote the equivalence class of $\mathcal{N}_{\mathbb{H}(P_i)}$ that $q^\#$ represents. We define a map $\phi : \textsc{Rch}(\mathbb{H}_{-1} \circ \mathbb{H}(P_i)) \to 2^{\textsc{Rch}(P'_{i'})}$ as follows:

$$\phi(q^\#_j) = \left\{ q'_j \in Q' \middle| \exists x \in \mathscr{E}(q^\#_j) \text{ s.t. } \delta'^\star(q_{i'}, x) = q'_j \right\} \tag{19}$$

We claim

$$\forall q^\#_j, q^\#_k \in Q^\# \Big(\big(q^\#_j \neq q^\#_k\big) \Rightarrow \big(\phi(q^\#_j) \bigcap \phi(q^\#_k) = \varnothing\big) \Big) \tag{\textbf{C2}}$$

Let $q'_\ell \in \phi(q^\#_j) \cap \phi(q^\#_k)$. Hence there exists $x_j \in \mathscr{E}(q^\#_j), x_k \in \mathscr{E}(q^\#_k)$ such that

$$q'_\ell = \delta'^\star(q_{i'}, x_j) = \delta'^\star(q_{i'}, x_k) \tag{20}$$

that

$$x_j \not\mathcal{N}_{\mathbb{H}(P_i)} x_k \Rightarrow \exists u \in \Sigma^\star \left(\frac{\mathbb{H}(P_i)(x_j u)}{\mathbb{H}(P_i)(x_j)} \neq \frac{\mathbb{H}(P_i)(x_k u)}{\mathbb{H}(P_i)(x_k)} \right) \tag{21}$$

But, $P'_{i'}$ perfectly encodes $\mathbb{H}(P_i)$ implying

$$\forall u = \sigma_1 \cdots \sigma_R \in \Sigma^\star \left(\frac{\mathbb{H}(P_i)(x_j u)}{\mathbb{H}(P_i)(x_j)} = \frac{\mathbb{H}(P_i)(x_k u)}{\mathbb{H}(P_i)(x_k)} \right.$$

$$\left. = \tilde{\pi}'(q'_\ell, \sigma_1) \prod_{r=1}^{R-1} \tilde{\pi}'(\delta'^\star(q_{i'}, \sigma_1 \cdots \sigma_r), \sigma_{r+1}) \right)$$

which contradicts Eq. (21).

Next we claim that

$$\forall q_j^{\#} \in \mathrm{RCH}(\mathbb{H}_{-1} \circ \mathbb{H}(P_i)),\ \mathrm{CARD}(\phi(q_j^{\#})) = 1 \qquad \textbf{(C3)}$$

Let $x_1, x_2 \in \mathscr{E}(q_j^{\#})$ such that

$$\left.\begin{aligned} \delta'^{\star}(q_{i'}, x_1) = q'_j \\ \delta'^{\star}(q_{i'}, x_2) = q'_k \end{aligned}\right\} \text{with } q'_j \neq q'_k \tag{22}$$

Therefore,

$$\begin{aligned} &\mathrm{CARD}(\phi(q_j^{\#})) > 1 \\ &\implies \sum_{q_k^{\#} \in \mathrm{RCH}(\mathbb{H}_{-1} \circ \mathbb{H}(P_i))} \mathrm{CARD}(\phi(q_k^{\#})) > \mathrm{CARD}(\mathrm{RCH}(\mathbb{H}_{-1} \circ \mathbb{H}(P_i))) \\ &\implies \mathrm{CARD}(\mathrm{RCH}(P'_{i'})) > \mathrm{CARD}(\mathrm{RCH}(\mathbb{H}_{-1} \circ \mathbb{H}(P_i))) \end{aligned}$$

which contradicts **C1** thus proving **C3**.

On account of **C2** and **C3**, let us define a bijective map $\widetilde{\phi} : \mathrm{RCH}(\mathbb{H}_{-1} \circ \mathbb{H}(P_i)) \to \mathrm{RCH}(P'_{i'})$ as $\widetilde{\phi}(q_j^{\#}) = \delta'^{\star}(q_{i'}, x), x \in \mathscr{E}(q_j^{\#})$. Then,

$$\begin{aligned} &\forall \sigma_k \in \Sigma, \forall q_j^{\#} \in \mathrm{RCH}(\mathbb{H}_{-1} \circ \mathbb{H}(P_i))), x \in \mathscr{E}(q_j^{\#}), \\ &\quad \tilde{\pi}^{\#}(q_j^{\#}, \sigma_k) = \frac{\mathbb{H}(P_i)(x\sigma_k)}{\mathbb{H}(P_i)(x)} = \tilde{\pi}^{\#}(\widetilde{\phi}(q_j^{\#}), \sigma_k) \qquad (23) \\ &\widetilde{\phi}(\delta^{\#}(q_j^{\#}, \sigma_k)) = \delta'^{\star}(q_{i'}, x\sigma_k) \\ &\qquad = \delta'(\delta'^{\star}(q_{i'}, \sigma_k), \sigma_k) = \delta'(\widetilde{\phi}(q_j^{\#}), \sigma_k) \qquad (24) \end{aligned}$$

which implies that $\mathbb{H}_{-1} \circ \mathbb{H}(P_i)$ and $P'_{i'}$ are isomorphic in the sense of Definition 3.6. This completes the proof. ❒

Theorem 3.4: *For a PFSA $P_i = (\mathscr{E}, \Sigma, \delta, E_i, \tilde{\pi})$, the function $\tilde{\pi} : Q \times \Sigma \to Q$ can be extended to $\tilde{\pi} : Q \times \Sigma^{\star} \to Q$ as:*

$$\forall q_j \in Q, \tau \in \Sigma^{\star}, \sigma \in \Sigma, \begin{cases} \tilde{\pi}(q_j, \varepsilon) = 1 \\ \tilde{\pi}(q_j, \sigma\tau) = \tilde{\pi}(q_j, \sigma)\tilde{\pi}(\delta(q_j, \sigma), \tau) \end{cases} \tag{25}$$

Proof: Let $\mathfrak{p} = \mathbb{H}(P_i)$. We note that that P_i perfectly encodes $\mathfrak{p}$ (See Lemma 3.1). It follows from Theorem 3.2 that

$$\forall q_j \in Q, \tilde{\pi}(q_j, \varepsilon) = \frac{\mathfrak{p}(x\varepsilon)}{\mathfrak{p}(x)} = \frac{\mathfrak{p}(x)}{\mathfrak{p}(x)} = 1 \text{ where } \delta^{\star}(q_i, x) = q_j$$

Similarly, for a string $\sigma\tau$ initiating from state q_j, where $\sigma \in \Sigma, \tau \in \Sigma^{\star}$, we have

$$\tilde{\pi}(q_j, \sigma\tau) = \frac{\mathfrak{p}(x\sigma\tau)}{\mathfrak{p}(x)} = \frac{\mathfrak{p}(x\sigma)}{\mathfrak{p}(x)} \times \frac{\mathfrak{p}(x\sigma\tau)}{\mathfrak{p}(x\sigma)} \tag{26}$$

We note that $\frac{\mathfrak{p}(x\sigma)}{\mathfrak{p}(x)} = \tilde{\pi}(q_j, \sigma)$. Also, $\delta^{\star}(q_i, x) = q_j$ implies $\delta(q_j, \sigma) = \delta^{\star}(q_i, x\sigma)$. Therefore, $\frac{\mathfrak{p}(x\sigma\tau)}{\mathfrak{p}(x\sigma)} = \tilde{\pi}(\delta(q_j, \sigma), \tau)$ and hence

$$\tilde{\pi}(q_j, \sigma\tau) = \tilde{\pi}(q_j, \sigma)\tilde{\pi}(\delta(q_j, \sigma), \tau)$$

This completes the proof. ❒

Theorem 3.5: *For a measure space $(\Sigma^{\omega}, \mathfrak{B}_{\Sigma}, \mathfrak{p})$,*

$$\mathbb{H} \circ \mathbb{H}_{-1}(\mathfrak{p}) = \mathfrak{p} \tag{27}$$

i.e., $\mathbb{H} \circ \mathbb{H}_{-1}$ is the identity map from $\mathscr{P}$ onto itself.

Proof: Let $\mathbb{H}_{-1}(\mathfrak{p}) = P_i = (Q, \Sigma, \delta, q_i, \tilde{\pi})$. We note P_i perfectly encodes $\mathfrak{p}$ (See Lemma 3.1). Let $\mathbb{H}(P_i) = \mathfrak{p}'$. We claim

$$\forall x \in \Sigma^{\star}, \mathfrak{p}(x) = \mathfrak{p}'(x)$$

The result is immediate for $|x| = 0$, i.e., $x = \epsilon$. For $|x| \geq 1$, we proceed by the method of induction. For $|x| = 1$, we note

$$\forall \sigma \in \Sigma, \mathfrak{p}'(\sigma) = \tilde{\pi}(q_i, \sigma) = \mathfrak{p}(\sigma) \text{ (Perfect Encoding)}$$

Next let us assume that $\forall x \in \Sigma^{\star},$ s.t. $|x| = r \in \mathbb{N}, \mathfrak{p}'(x) = \mathfrak{p}(x)$. Since $\forall x \in \Sigma^{\star}$ with $|x| = r \in \mathbb{N}$, it follows that

$$\begin{aligned} \mathfrak{p}'(x\sigma) &= \mathfrak{p}'(x)\tilde{\pi}(q_j, \sigma) \text{ where } \delta^{\star}(q_i, x) = q_j \\ &= \mathfrak{p}(x)\tilde{\pi}(q_j, \sigma) = \mathfrak{p}(x\sigma) \end{aligned}$$

This completes the proof. ❒

4. Metrization of the Space $\mathscr{P}$ of Probability Measures on $\mathfrak{B}_{\Sigma}$

Metrization of $\mathscr{P}$ is important for differentiating physical processes modeled as dynamical systems evolving probabilistically on discrete state spaces of finite cardinality. In this section, we introduce two metric families, each of which captures a different aspect of such dynamical behavior and can be combined to form physically meaningful and useful metrics for system analysis and design.

Definition 4.1: *Given two probability measures $\mathfrak{p}_1, \mathfrak{p}_2$ on the σ-algebra $\mathfrak{B}_{\Sigma}$ and a parameter $s \in [1, \infty]$, the function $d_s : \mathscr{P} \times \mathscr{P} \to [0, 1]$ is defined as follows:*

$$d_s(\mathfrak{p}_1, \mathfrak{p}_2) = \sup_{x \in \Sigma^{\star}} \left(\sum_{j=1}^{|\Sigma|} \left| \frac{\mathfrak{p}_1(x\sigma_j)}{\mathfrak{p}_1(x)} - \frac{\mathfrak{p}_2(x\sigma_j)}{\mathfrak{p}_2(x)} \right|^s \right)^{1/s} \forall s \in [1, \infty) \tag{28a}$$

$$d_{\infty}(\mathfrak{p}_1, \mathfrak{p}_2) = \sup_{x \in \Sigma^{\star}} \max_{\sigma \in \Sigma} \left| \frac{\mathfrak{p}_1(x\sigma_j)}{\mathfrak{p}_1(x)} - \frac{\mathfrak{p}_2(x\sigma_j)}{\mathfrak{p}_2(x)} \right| \tag{28b}$$

Theorem 4.1: *The space $\mathscr{P}$ of all probability measures on $\mathfrak{B}_{\Sigma}$ is d_s-metrizable for $s \in [1, \infty]$.*

Proof: Strict positivity and symmetry properties of a metric follow directly from Definition 4.1. Validity of the remaining property of triangular inequality follows by application of Minkowski inequality [17]. ❒

Definition 4.2: *Let $\mathcal{M}$ be a right invariant equivalence relation on $\Sigma^{\star}$ with the i^{th} equivalence class of $\mathcal{M}$ be denoted as $\mathcal{M}^i$, $i \in \mathcal{I}$, where $\mathcal{I}$ is an arbitrary index set. Let $\mathfrak{p}$ be a probability measure on the σ-algebra $\mathfrak{B}_{\Sigma}$ inducing the probabilistic Nerode equivalence $\mathcal{N}_{\mathfrak{p}}$ on $\Sigma^{\star}$ with the j^{th} equivalence class of $\mathcal{N}_{\mathfrak{p}}$ denoted as $\mathcal{N}_{\mathfrak{p}}^j$, $j \in \mathcal{J}$, where $\mathcal{J}$ is an index set distinct from $\mathcal{I}$. Then, the map $\Omega_{\mathcal{M}} : \mathscr{P} \longrightarrow [0,1]^{\mathrm{CARD}(\mathcal{I})} \times [0,1]^{\mathrm{CARD}(\mathcal{J})}$ is defined as*

$$\Omega_{\mathcal{M}}(\mathfrak{p})\Big|_{ij} = \sum_{x \in \mathcal{M}^i \cap \mathcal{N}_{\mathfrak{p}}^j} \mathfrak{p}(x)$$

Definition 4.3: *Let $\mathfrak{p}_1, \mathfrak{p}_2$ be two probability measures on the σ-algebra $\mathfrak{B}_\Sigma$. Then, the function $d_F : \mathscr{P} \times \mathscr{P} \to [0,1]$ is defined as follows:*

$$d_F(\mathfrak{p}_1, \mathfrak{p}_2) = \left\|\Omega_{\mathcal{N}_{\mathfrak{p}_2}}(\mathfrak{p}_1) - \Omega_{\mathcal{N}_{\mathfrak{p}_1}}(\mathfrak{p}_2)\right\|_F \tag{29}$$

where $\left\|\Theta\right\|_F = \sqrt{Trace[\Theta^H\Theta]}$ is the Frobenius norm of the operator Θ, and Θ^H is the Hermitian of Θ.

Definition 4.3 implies that if $\mathcal{I}$ and $\mathcal{J}$ are the index sets corresponding to $\mathcal{N}_{\mathfrak{p}_1}$ and $\mathcal{N}_{\mathfrak{p}_2}$ respectively, then $\Omega_{\mathcal{N}_{\mathfrak{p}_1}}(\mathfrak{p}_2) \in [0,1]^{\text{CARD}(\mathcal{I})} \times [0,1]^{\text{CARD}(\mathcal{J})}$ and $\Omega_{\mathcal{N}_{\mathfrak{p}_2}}(\mathfrak{p}_1) \in [0,1]^{\text{CARD}(\mathcal{J})} \times [0,1]^{\text{CARD}(\mathcal{I})}$.

Theorem 4.2: *The function d_F is a pseudometric on the space $\mathscr{P}$ of probability measures.*

Proof: The Frobenius norm on a probability space satisfies the metric properties except strict positivity because of the almost sure property of a probability measure. ❐

Theorem 4.3: *For $\forall\alpha \in [0,1)$ and $\forall s \in [1,\infty]$, the parametized function $\mu_{\alpha,s} \triangleq \alpha d_F + (1-\alpha)d_s$ is a metric on $\mathscr{P}$.*

Proof: Following Theorems 4.1 and 4.2, d_s is a metric for $s \in [1,\infty]$ and d_F is a pseudometric on $\mathscr{P}$. Non-negativity, finiteness, symmetry and sub-additivity of $\mu_{\alpha,s}$ follow from the respective properties of d_F and d_s. Strict positivity of $\mu_{\alpha,s}$ on $\alpha \in [0,1)$ is established below.

$$\mu_{\alpha,s}(\mathfrak{p}_1, \mathfrak{p}_2) = 0 \Rightarrow (1-\alpha)d_s(\mathfrak{p}_1, \mathfrak{p}_2) = 0 \Rightarrow \mathfrak{p}_1 = \mathfrak{p}_2 \tag{30}$$

❐

Remark 4.1: *If two physical processes are modeled as discrete-event dynamical systems, then the respective probabilistic language generators can be associated with probability measures $\mathfrak{p}_1$ and $\mathfrak{p}_2$. The metric $d_s(\mathfrak{p}_1, \mathfrak{p}_2)$ is related to the production of single symbols as arbitrary strings and hence captures the difference in short term dynamic evolution. In contrast, the pseudometric d_F is related to generation of all possible strings and therefore captures the difference in long term behavior of the physical processes. The metric $\mu_{\alpha,s}$ thus captures the both short-term and long-term behavior with respective relative weights of $1-\alpha$ and α.*

Definition 4.4: *The metric $\mu_{\alpha,s}$ on $\mathscr{P}$ for $\alpha \in [0,1), s \in [1,\infty]$, induces a function $\nu_{\alpha,s}$ on $\mathscr{A} \times \mathscr{A}$ as follows:*

$$\forall P_i, P'_{i'} \in \mathscr{A}, \nu_{\alpha,s}(P_i, P'_{i'}) = \mu_{\alpha,s}(\mathbb{H}(P_i), \mathbb{H}(P'_{i'})) \tag{31}$$

Corollary 4.1: (to Theorem 4.3) *The function $\nu_{\alpha,s}$ in Definition 4.4 for $\alpha \in [0,1)$ and $s \in [1,\infty]$ is a pseudometric on $\mathscr{A}$. Specifically, the following condition holds:*

$$\nu_{\alpha,s}(P_i, \mathbb{H}_{-1} \circ \mathbb{H}(P_i)) = 0 \tag{32}$$

Proof: Following Theorem 3.5,

$$\nu_{\alpha,s}(P_i, \mathbb{H}_{-1} \circ \mathbb{H}(P_i)) = \mu_{\alpha,s}(\mathbb{H}(P_i), \mathbb{H} \circ \mathbb{H}_{-1} \circ \mathbb{H}(P_i))$$
$$= \mu_{\alpha,s}(\mathbb{H}(P_i), \left(\mathbb{H} \circ \mathbb{H}_{-1}\right) \circ \mathbb{H}(P_i)) = \mu_{\alpha,s}(\mathbb{H}(P_i), \mathbb{H}(P_i)) = 0$$

❐

Remark 4.2: *Corollary 4.1 can be physically interpreted to imply that the metric family $\nu_{\alpha,s}$ does not diffrentiate between different realizations of the same probability measure. Thus when comparing two probabilistic finite state machines, we need not concern ourselves with whether the machines are represented in their minimal realizations; the distance between two non-minimal realizations of the same PFSA is always zero. However this implies that $\nu_{\alpha,s}$ only qualifies as a pseudo-metric on $\mathscr{A}$.*

A. Explicit Computation of the Pseudometric ν for PFSA

The pseudometric $\nu_{\alpha,s}$ is computed explicitly for pairs of PFSA over the same alphabet. Before proceeding to the general case, $\nu_{\alpha,s}$ is computed for the special case, where the pair of PFSA have identical state sets, initial states and transition maps.

Lemma 4.1: *Given two PFSA $P_i^1 = (Q,\Sigma,\delta,q_i,\tilde{\pi}^1)$, $P_i^2 = (Q,\Sigma,\delta,q_i,\tilde{\pi}^2)$, and $\sigma \in \Sigma$, the steps for computation of $\nu_{0,s}(P_i^1, P_i^2)$ are:*

$$\text{Set } \Delta(q_j) = \tilde{\pi}^1(q_j,\sigma) - \tilde{\pi}^2(q_j,\sigma)$$
$$\text{Then, } \nu_{0,s}(P_i^1, P_i^2) = \max_{q_j \in Q} \left\|\Delta(q_j)\right\|_s$$

Proof: Let $\frac{\mathbb{H}(P_i^2)(x\sigma)}{\mathbb{H}(P_i^2)(x)}$ denote a $|\Sigma|$-dimensional vector-valued function, where $\sigma \in \Sigma$. For proof of the lemma, it suffices to show that the following relation holds:

$$\sup_{x \in \Sigma^\star} d_s\left(\frac{\mathbb{H}(P_i^1)(x\sigma)}{\mathbb{H}(P_i^1)(x)}, \frac{\mathbb{H}(P_i^2)(x\sigma)}{\mathbb{H}(P_i^2)(x)}\right) = \max_{q_j \in Q} \left\|\Delta(q_j)\right\|_s$$

Since P_i^1 perfectly encodes $\mathbb{H}(P_i^1)$, it follows that $\left(\forall x, y \in \Sigma^\star, \delta(q_i,x) = \delta(q_i,y)\right)$ implies

$$\frac{\mathbb{H}(P_i^1)(x\sigma_k)}{\mathbb{H}(P_i^1)(x)} = \tilde{\pi}^1(q_j,\sigma_k) = \frac{\mathbb{H}(P_i^1)(y\sigma)}{\mathbb{H}(P_i^1)(y)}$$

where $\delta(q_i,x) = q_j$. Similar argument holds for $\mathbb{H}(P_i^2)$. Hence, it follows that for computing $\nu_{0,s}(P_i^1, P_i^2)$, only one string needs to be considered for each state $q_j \in Q$. That is,

$$\nu_{0,s}(P_i^1, P_i^2) = \max_{x:\delta(q_i,x)=q_j} \left\|\frac{\mathbb{H}(P_i^1)(x\sigma_k)}{\mathbb{H}(P_i^1)(x)} - \frac{\mathbb{H}(P_i^2)(x\sigma_k)}{\mathbb{H}(P_i^2)(x)}\right\|_s$$
$$= \max_{x:\delta(q_i,x)=q_j} \left\|\tilde{\pi}^1(q_j,\sigma_k) - \tilde{\pi}^2(q_j,\sigma_k)\right\|_s = \max_{q_j \in Q} \left\|\Delta(q_j)\right\|_s$$

❐

Lemma 4.2: *Let $\wp_1, \wp_2$ be the stable probability distributions for PFSA $P_i^1 = (Q,\Sigma,\delta,q_i,\tilde{\pi}^1)$ and $P_i^2 = (Q,\Sigma,\delta,q_i,\tilde{\pi}^2)$ respectively. Then,*

$$\lim_{\alpha \to 1} \nu_{\alpha,s}(P_i^1, P_i^2) = d_2\left(\wp_1, \wp_2\right)$$

Proof: Since P_i^1 and P_i^2 have the same initial state and state transition maps,

$$\mathcal{N}^j_{\mathbb{H}(P_i^1)} \bigcap \mathcal{N}^k_{\mathbb{H}(P_i^2)} = \begin{cases} \varnothing & \text{if } j \neq k \\ \mathcal{N}^j_{\mathbb{H}(P_i^1)} = \mathcal{N}^k_{\mathbb{H}(P_i^2)} & \text{otherwise} \end{cases}$$

where $\mathcal{N}^j_{\mathbb{H}(P_i^1)}$ and $\mathcal{N}^k_{\mathbb{H}(P_i^2)}$ are the j^{th} and k^{th} equivalence classes (i.e., states q_j and q_k) for P_i^1, P_i^2, respectively. The result follows from Definition 4.3 and Corollary 4.4 and noting that

$$\Omega_{\mathcal{N}_{\mathbb{H}(P_i^2)}}(\mathbb{H}(P_i^1))\Big|_{jk} = \begin{cases} \sum\limits_{x:\delta(q_i,x)=q_j} \mathfrak{p}(x) = \wp_1\Big|_j & \text{if } j = k \\ 0 & \text{otherwise} \end{cases}$$

❒

Theorem 4.4: *Given two PFSA $P_i^1 = (Q, \Sigma, \delta, q_i, \tilde{\pi}^1)$ and $P_i^2 = (Q, \Sigma, \delta, q_i, \tilde{\pi}^2)$, the pseudometric $\boldsymbol{\nu}_{\alpha,s}(P_i^1, P_i^2)$ can be computed explicitly for $\alpha \in [0, 1)$ and $s \in [1, \infty]$ as:*

$$\boldsymbol{\nu}_{\alpha,s}(P_i^1, P_i^2) = \alpha \lim_{\alpha \to 1} \boldsymbol{\nu}_{\alpha,s}(P_i^1, P_i^2) + (1 - \alpha)\boldsymbol{\nu}_{0,s}(P_i^1, P_i^2) \tag{33}$$

Proof: The result follows from Theorem 4.3 and Corollary 4.4. ❒

The algorithm for computation of the the pseudometric $\boldsymbol{\nu}$ is presented below.

Algorithm 2: Computation of $\boldsymbol{\nu}_{\alpha,s}(P_i, P'_{i'})$

input : $P_i, P'_{i'}, s, \alpha$
output: $\boldsymbol{\nu}_{\alpha,s}(P_i, P'_{i'})$
begin
  Compute $\mathbf{P}_{12} = (\overline{\mathbf{Q}}, \Sigma, \overline{\boldsymbol{\delta}}, \tilde{\boldsymbol{\pi}}_{12}) = P_i \otimes P'_{i'}$;
  Compute $\mathbf{P}_{21} = (\overline{\mathbf{Q}}, \Sigma, \overline{\boldsymbol{\delta}}, \tilde{\boldsymbol{\pi}}_{21}) = P'_{i'} \otimes P_i$;
  for $j = 1$ ***to*** CARD($\overline{\mathbf{Q}}$) **do**
    $\Delta(j) = \left\|\tilde{\boldsymbol{\pi}}_{12}(q_j, \sigma_k) - \tilde{\boldsymbol{\pi}}_{21}(q_j, \sigma_k)\right\|_s$;
  endfor
  $\nu_{0,s}(P_i, P'_{i'}) = \max_j \Delta(j)$;
  Compute $\wp_{12}$; `/* State Prob. for` $\mathbf{P}_{12}$ `(Def. 2.3) */`
  Compute $\wp_{21}$; `/* State Prob. for` $\mathbf{P}_{21}$ `(Def. 2.3) */`
  Compute $d = \|\wp_{12} - \wp_{21}\|_2$;
  Compute $\boldsymbol{\nu}_{\alpha,s}(P_i, P'_{i'}) = \alpha d + (1 - \alpha)\boldsymbol{\nu}_{0,s}(P_i, P'_{i'})$;
end

To extend the approach presented in Lemma 4.1 to arbitrary pairs of PFSA, we need to define the synchronous composition of a pair of PFSA.

Definition 4.5: *The binary operation of synchronous composition of PFSA, denoted as $\otimes : \mathscr{P} \times \mathscr{P} \longrightarrow \mathscr{P}$, is defined as follows:*

$$\text{Let } \begin{cases} P_i = (Q, \Sigma, \delta, q_i, \tilde{\pi}) \\ G_{i'} = (Q', \Sigma, \delta', q'_{i'}, \tilde{\pi}') \end{cases}$$

Then, $P_i \otimes G_{i'} = (Q \times Q', \Sigma, \delta^{\otimes}, (q_i, q'_{i'}), \tilde{\pi}^{\otimes})$, where

$$\delta^{\otimes}((q_j, q'_{k'}), \sigma) = \begin{cases} \big(\delta(q_j, \sigma), \delta'(q'_{j'}, \sigma)\big), & \textit{if } \delta(q_j, \sigma) \textit{ and } \\ & \delta'(q'_{j'}, \sigma) \textit{ are defined.} \\ \text{Undefined} & \textit{otherwise} \end{cases} \tag{34}$$

$$\tilde{\pi}^{\otimes}((q_j, q'_{k'}), \sigma) = \tilde{\pi}(q_j, \sigma) \tag{35}$$

Remark 4.3: *Synchronous composition for PFSA is not commutative, i.e., for an arbitrary pair P_i and $G_{i'}$,*

$$P_i \otimes G_{i'} \neq G_{i'} \otimes P_i \tag{36}$$

Synchronous composition of PFSA is associative, i.e.,

$$\forall P^1_{i1}, P^2_{i2}, P^3_{i3} \in \mathscr{P}, \left(P^1_{i1} \otimes P^2_{i2}\right) \otimes P^3_{i3} = P^1_{i1} \otimes \left(P^2_{i2} \otimes P^3_{i3}\right)$$

Theorem 4.5: *For a pair of PFSA P_i and $G_{i'}$ over the same alphabet,*

$$\mathbb{H}(P_i) = \mathbb{H}(P_i \otimes G_{i'}) \tag{37}$$

Proof: Let $\mathfrak{p} = \mathbb{H}(P_i)$ and $\mathfrak{p}' = \mathbb{H}(P_i \otimes G_{i'})$. It suffices to show that

$$\forall x \in \Sigma^\star, \mathfrak{p}(x) = \mathfrak{p}'(x) \tag{C4}$$

For $|x| = 0$, i.e., $x = \epsilon$, the result is immediate. For $|x| \geqq 1$, we use the method of induction. Since P_i perfectly encodes $\mathbb{H}(P_i)$,

$$\begin{aligned} \forall \sigma \in \Sigma, \mathfrak{p}(\sigma) &= \tilde{\pi}(q_i, \sigma) \\ &= \tilde{\pi}^{\otimes}((q_i, q'_{i'}), \sigma) = \mathfrak{p}'(\sigma) \end{aligned}$$

Hence **C4** is true for $|x| \leqq 1$.

With the induction hypothesis

$$\forall x \in \Sigma^\star, \text{ s.t. } |x| = r \in \mathbb{N}, \mathfrak{p}(x) = \mathfrak{p}'(x) \tag{38}$$

we proceed with an arbitrary $\sigma \in \Sigma$ to yield

$$\begin{aligned} &\mathfrak{p}(x\sigma) = \mathfrak{p}(x)\tilde{\pi}(q_j, \sigma) \text{ where } \delta(q_i, x) = q_j \\ &= \mathfrak{p}'(x)\tilde{\pi}^{\otimes}((q_j, q'_{j'}), \sigma) \text{ where } \delta^{\otimes\star}((q_i, q'_{i'}), x) = (q_j, q'_{j'}) \\ &= \mathfrak{p}'(x\sigma) \end{aligned}$$

This completes the proof. ❒

Theorem 4.6: *Given a pair of PFSA $P_i, P'_{i'}$ and an arbitrary parameter $s \in [1, \infty]$, Algorithm 2 computes $\boldsymbol{\nu}_{\alpha,s}(P_i, P'_{i'})$ for $\alpha \in [0, 1), s \in [1, \infty]$.*

Proof: By Theorem 4.5, $\forall \alpha \in [0, 1), s \in [1, \infty]$,

$$\boldsymbol{\nu}_{\alpha,s}(P_i, P'_{i'}) = \boldsymbol{\nu}_{\alpha,s}(P_i \otimes P'_{i'}, P'_{i'} \otimes P_i) \tag{39}$$

Since $P_i \otimes P'_{i'}$ and $P'_{i'} \otimes P_i$ have the same state sets, initial states and transition maps (See Definition 4.5), correctness of Algorithm 2 follows from Lemmas 4.1 and 4.2. ❒

Example 4.1: *The theoretical results of Section 4 are illustrated with a numerical example. The following PFSA are considered:*

$$P^1_{q_1} = (\{q_1, q_2, q_3\}, \{0, 1\}, \delta^1, q_1, \tilde{\pi}^1) \tag{40}$$

$$P^2_{q_A} = (\{q_A, q_B\}, \{0, 1\}, \delta^2, q_A, \tilde{\pi}^2) \tag{41}$$

as shown in Fig. 6 and Fig. 7, respectively, and Fig. 8 illustrates the computed compositions $P^1_{q_1} \otimes P^2_{q_A}$ (above) and $P^2_{q_A} \otimes P^1_{q_1}$ (below).

Following Algorithm 2, we have

$$\Delta_s = \begin{bmatrix} \|0.4 - 0.9 & 0.6 - 0.1\|_s \\ \|0.2 - 0.9 & 0.8 - 0.1\|_s \\ \|0.3 - 0.9 & 0.7 - 0.1\|_s \\ \|0.3 - 0.7 & 0.7 - 0.3\|_s \\ \|0.4 - 0.7 & 0.6 - 0.3\|_s \\ \|0.2 - 0.7 & 0.8 - 0.3\|_s \end{bmatrix} \tag{42}$$

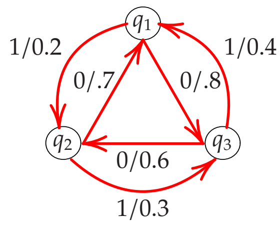


Fig. 6. PFSA P^1

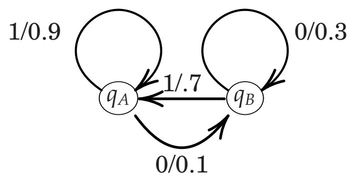


Fig. 7. PFSA P^2

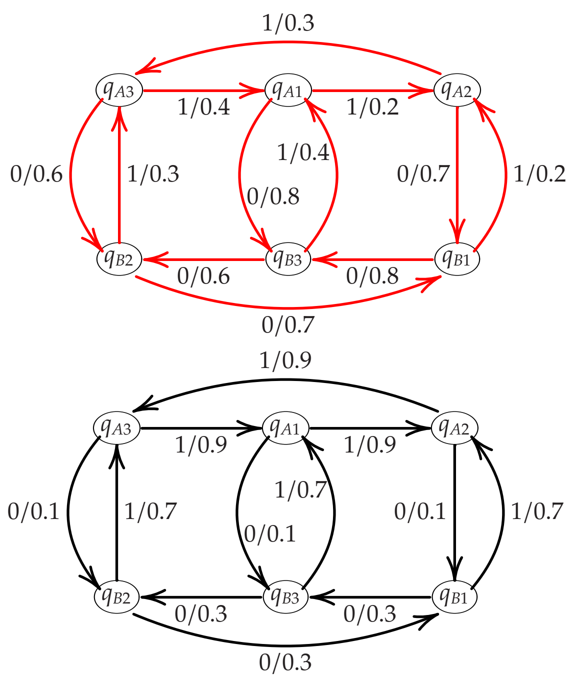


Fig. 8. $P^1 \otimes P^2$ (above) and $P^2 \otimes P^1$ (below)

As an illustration, we set $s = \infty$. Hence,

$$\Delta_\infty = \begin{bmatrix} 0.5 & 0.7 & 0.6 & 0.4 & 0.3 & 0.5 \end{bmatrix}^T \tag{43}$$

$$\implies \nu_{0,\infty}(P^1_{q_1}, P^2_{q_A}) = max(\Delta_\infty) = 0.7 \tag{44}$$

The final state probabilities are computed to be

$$\wp_{P^1 \otimes P^2} = [0.15\ 0.07\ 0.09\ 0.2\ 0.22\ 0.27] \tag{45}$$

$$\wp_{P^2 \otimes P^1} = [0.29\ 0.29\ 0.29\ 0.043\ 0.043\ 0.043] \tag{46}$$

For $\alpha = 0.5$, we have

$$\begin{aligned} \nu_{0.5,\infty}(P^1_{q_1}, P^2_{q_A}) &= 0.5 d_2(\wp_{P^1 \otimes P^2}, \wp_{P^2 \otimes P^1}) + 0.5 \times 0.7 \\ &= 0.4599 \end{aligned}$$

The pseudonorm $\nu_{0,\infty}(P^1_{q_1}, P^2_{q_A}) = 0.7$ is interpreted as follows. There exists a string $x \in \Sigma^\star$ and an event $\sigma \in \Sigma$ such that probability of occurrence of σ, given that x has already occurred, is 70% more in one system compared to the other. Also, the occurrence probability of any event, given an arbitrary string has already occurred, is different by no more than 70% for the two systems. The composition $P^1_{q_1} \otimes P^2_{q_A}$ shown in the upper part of Fig. 8 is an encoding of the measure $\mathbb{H}(P^1_i)$ and hence is a non-minimal realization of P^1_i, while the composition $P^2_{q_A} \otimes P^1_{q_1}$ shown in the lower part of Fig. 8 encodes $\mathbb{H}(P^2_{i'})$ and therefore is a non-minimal realization of $P^2_{i'}$. Although the structures of the two compositions are identical in a graph-theoretic sense (i.e. there is a graph isomorphism between the compositions), they represent very different probability distributions on $\mathfrak{B}_\Sigma$.

5. Model Order Reduction for PFSA

This section investigates the possibility of encoding an arbitrary probability distribution on $\mathfrak{B}_\Sigma$ by a PFSA with a pre-specified graph structure. As expected, such encodings will not always be perfect. However, we will show that the error can be rigorously computed and hence is useful for very close approximation of large PFSA models by smaller models.

Definition 5.1: *The binary operation of projective composition $\overrightarrow{\otimes} : \mathscr{P} \times \mathscr{P} \rightarrow \mathscr{P}$ is defined as follows:*

$$\text{Let} \begin{cases} P_i = (Q, \Sigma, \delta, q_i, \tilde{\pi}) \\ G_{i'} = (Q', \Sigma, \delta', q'_{i'}, \tilde{\pi}') \\ G_{i'} \otimes P_i = (Q' \times Q, \Sigma, \delta^\otimes, (q'_{i'}, q_i), \tilde{\pi}^\otimes) \end{cases}$$

For notational simplicity set $\forall q_j \in Q$ *and* $\forall q'_{k'} \in Q'$,

$$\vartheta(q'_{k'}, q_j) = \sum_{\substack{x : \delta^{\otimes\star}((q'_{i'}, q_i), x) \\ = (q'_{k'}, q_j)}} \Big(\mathbb{H}(G_{i'})(x) \Big)$$

Then, $G_{i'} \overrightarrow{\otimes} P_i = (Q, \Sigma, \delta, q_i, \tilde{\pi}^{\overrightarrow{\otimes}})$ s.t.

$$\tilde{\pi}^{\overrightarrow{\otimes}}(q_j, \sigma) = \frac{\sum_{q'_{k'} \in Q'} \vartheta(q'_{k'}, q_j) \tilde{\pi}^\otimes((q'_{k'}, q_j), \sigma)}{\sum_{q'_{k'} \in Q'} \vartheta(q'_{k'}, q_j)} \tag{47}$$

Theorem 5.1: *For PFSA P_i, G_j, H_k over the same alphabet,*

1. $P_i \overrightarrow{\otimes} \left(G_j \overrightarrow{\otimes} H_k \right) = P_i \overrightarrow{\otimes} H_k$
2. $\left(P_i \overrightarrow{\otimes} G_j \right) \overrightarrow{\otimes} H_k \neq P_i \overrightarrow{\otimes} \left(G_j \overrightarrow{\otimes} H_k \right)$ *(Non-associative)*
3. $P_i \overrightarrow{\otimes} G_j \neq G_j \overrightarrow{\otimes} P_i$ *(Non-commutative)*

Proof: The results follow from Definition 5.1. ❒

We justify the nomenclature "projective" composition in the following theorem.

Theorem 5.2: *For arbitrary PFSA P_i and $G_{i'}$ over the same alphabet,*

$$\left(G_{i'} \overrightarrow{\otimes} P_i \right) \overrightarrow{\otimes} P_i = G_{i'} \overrightarrow{\otimes} P_i \tag{48a}$$

Proof: Let $P_i = (Q, \Sigma, \delta, q_i, \tilde{\pi})$. Definition 5.1 implies that $\left(G_{i'} \overrightarrow{\otimes} P_i \right) = (Q, \Sigma, \delta, q_i, \tilde{\pi}^\ddagger)$ for $\tilde{\pi}^\ddagger$ computed as specified in Eq. (47). It further follows from Definition 5.1, that $\left(G_{i'} \overrightarrow{\otimes} P_i \right) \overrightarrow{\otimes} P_i = (Q, \Sigma, \delta, q_i, \tilde{\pi}^{\overrightarrow{\otimes}})$, i.e. $\left(G_{i'} \overrightarrow{\otimes} P_i \right) \overrightarrow{\otimes} P_i$ and $G_{i'} \overrightarrow{\otimes} P_i$ have the same state set, initial state and state transition maps. Thus, it suffices to show that

$$\forall q_j \in Q, \sigma \in \Sigma, \tilde{\pi}^\ddagger(q_j, \sigma) = \tilde{\pi}^{\overrightarrow{\otimes}}(q_j, \sigma) \tag{49}$$

Considering the probabilistic synchronous composition $\left(G_{i'} \overrightarrow{\otimes} P_i \right) \otimes P_i = (Q \times Q, \Sigma, \delta^\otimes, (q_i, q_i), \tilde{\pi}^\otimes)$ (See Definition 4.5),

$$\forall x \in \Sigma^\star, \ \delta^{\otimes\star}((q_i, q_i), x) = (q_j, q_j), \text{ for some } q_j \in Q$$

It follows that, for $q_k \neq q_j$,

$$\vartheta(q_k, q_j) = \sum_{\substack{x:\delta^{\otimes\star}((q_i,q_i),x)\\=(q_k,q_j)}} \left(\mathbb{H}(G_{i'}\overrightarrow{\otimes}P_i)(x)\right) = 0 \tag{50}$$

Finally we conclude $\forall q_j \in Q, \sigma \in \Sigma$,

$$\begin{aligned}\tilde{\pi}^{\overrightarrow{\otimes}}(q_j,\sigma) &= \frac{\sum_{q_k\in Q}\vartheta(q_k,q_j)\tilde{\pi}^{\otimes}((q_k,q_j),\sigma)}{\sum_{q_k\in Q}\vartheta(q_k,q_j)}\\ &= \frac{\vartheta(q_j,q_j)\tilde{\pi}^{\otimes}((q_j,q_j),\sigma)}{\vartheta(q_j,q_j)}\\ &= \tilde{\pi}^{\otimes}((q_j,q_j),\sigma)\\ &= \tilde{\pi}^{\ddagger}(q_j,\sigma) \text{ (See Definition 4.5)}\end{aligned} \tag{51}$$

This completes the proof. ❐

Projective composition preserves the projected distribution which is defined next.

Definition 5.2: ***(Projected Distribution:)*** *The projected distribution $\wp \in [0,1]^{\textsc{NumStates}(P_i)}$ of an arbitrary PFSA $G_{i'}$ with respect to a given PFSA P_i is defined by the map $[\![\cdot]\!]_{P_i} : \mathscr{A} \rightarrow [0,1]^{\textsc{NumStates}(P_i)}$ as follows:*

$$[\![G_{i'}]\!]_{P_i} = \wp \in [0,1]^{\textsc{NumStates}(P_i)},$$

suchthat if $\mathcal{N}^j$ is the j^{th} equivalence class (i.e. the j$^{\text{th}}$ state) of P_i, then $\sum_{x\in\mathcal{N}^j}\mathbb{H}(G_{i'})(x) = \wp_j$

We note $[\![G_{i'}]\!]_{P_i}$ is a probability vector, i.e.,

$$\sum_{j=1}^{\textsc{NumStates}(P_i)} [\![G_{i'}]\!]_{P_i}\Big|_j = \sum_{x\in\Sigma^\star}\mathbb{H}(G_{i'})(x) = 1 \tag{52}$$

Theorem 5.3: ***(Projected Distribution Invariance:)*** *For two arbitrary PFSA P_i and $G_{i'}$ over the same alphabet,*

$$[\![G_{i'}]\!]_{P_i} = [\![G_{i'}\overrightarrow{\otimes}P_i]\!]_{P_i}$$

Proof: Let $P_i = (Q,\Sigma,\delta,q_i,\tilde{\pi})$ and $G_{i'} = (Q',\Sigma,\delta',q'_{i'},\tilde{\pi}')$. It follows that $G_{i'}\overrightarrow{\otimes}P_i = (Q,\Sigma,\delta,q_i,\tilde{\pi}^{\overrightarrow{\otimes}})$, where $\tilde{\pi}^{\overrightarrow{\otimes}}$ is as computed in Definition 5.1. Using the same notation as in Definition 5.1, we have $\forall\sigma\in\Sigma$,

$$\begin{aligned}\sum_{x:\delta^\star(q_i,x)=q_j}\mathbb{H}(G_{i'})(x\sigma) &= \sum_{q'_{k'}\in Q'}\vartheta(q'_{k'},q_j)\tilde{\pi}'(q'_{k'},\sigma)\\ &= \sum_{q'_{k'}\in Q'}\vartheta(q'_{k'},q_j)\left\{\frac{\sum_{q'_{k'}\in Q'}\vartheta(q'_{k'},q_j)\tilde{\pi}'(q'_{k'},\sigma)}{\sum_{q'_{k'}\in Q'}\vartheta(q'_{k'},q_j)}\right\}\\ &= \left\{\sum_{q'_{k'}\in Q'}\vartheta(q'_{k'},q_j)\right\}\tilde{\pi}^{\overrightarrow{\otimes}}(q_j,\sigma)\end{aligned} \tag{53}$$

Since $[\![G_{i'}]\!]_{P_i}\Big|_j = \sum_{x:\delta^\star(q_i,x)=q_j}\mathbb{H}(G_{i'})(x) = \sum_{q'_{k'}\in Q'}\vartheta(q'_{k'},q_j)$, it follows that $\forall\sigma\in\Sigma$,

$$\frac{\sum_{x:\delta^\star(q_i,x)=q_j}\mathbb{H}(G_{i'})(x\sigma)}{[\![G_{i'}]\!]_{P_i}\Big|_j} = \tilde{\pi}^{\overrightarrow{\otimes}}(q_j,\sigma)$$

$$\Rightarrow \sum_{\sigma:\delta(q_j,\sigma)=q_\ell}\frac{\sum_{x:\delta^\star(q_i,x)=q_j}\mathbb{H}(G_{i'})(x\sigma)}{[\![G_{i'}]\!]_{P_i}\Big|_j} = \sum_{\sigma:\delta(q_j,\sigma)=q_\ell}\tilde{\pi}^{\overrightarrow{\otimes}}(q_j,\sigma)$$

$$\Rightarrow \frac{1}{[\![G_{i'}]\!]_{P_i}\Big|_j}\mathbb{H}(G_{i'})(x\Sigma_{j\ell}) = \pi^{\overrightarrow{\otimes}}(q_j,q_\ell) \tag{54}$$

where $\Sigma_{j\ell} \subsetneqq \Sigma$ such that $\sigma\in\Sigma_{j\ell}\Rightarrow\delta(q_j,\sigma)=q_\ell$ and $\pi^{\overrightarrow{\otimes}}(q_j,q_\ell)$ is the $j\ell^{th}$ element of the stochastic state transition matrix $\Pi^{\overrightarrow{\otimes}}$ corresponding to the PFSA $G_{i'}\overrightarrow{\otimes}P_i$. It follows from Eq. (54), that

$$\begin{aligned}\sum_{q_j\in Q}\mathbb{H}(G_{i'})(x\Sigma_{j\ell}) &= \sum_{q_j\in Q}[\![G_{i'}]\!]_{P_i}\Big|_j\pi^{\overrightarrow{\otimes}}(q_j,q_\ell)\\ \Rightarrow [\![G_{i'}]\!]_{P_i}\Big|_\ell &= \sum_{q_j\in Q}[\![G_{i'}]\!]_{P_i}\Big|_j\pi^{\overrightarrow{\otimes}}(q_j,q_\ell)\end{aligned} \tag{55}$$

It follows that $[\![G_{i'}]\!]_{P_i}$ satisfies the vector equation

$$[\![G_{i'}]\!]_{P_i} = [\![G_{i'}]\!]_{P_i}\Pi^{\overrightarrow{\otimes}} \tag{56}$$

We note that $[\![G_{i'}\overrightarrow{\otimes}P_i]\!]_{P_i}$ is the stable probability distribution of the PFSA $G_{i'}\overrightarrow{\otimes}P_i$ and hence, we have

$$[\![G_{i'}\overrightarrow{\otimes}P_i]\!]_{P_i} = [\![G_{i'}\overrightarrow{\otimes}P_i]\!]_{P_i}\Pi^{\overrightarrow{\otimes}} \tag{57}$$

In general, a stochastic matrix may have more than one eigenvector corresponding to unity eigenvalue [18]. However, as per our definition of PFSA (See Definition 2.2), the initial state is explicitly specified. It follows that the right hand side of Eq.(53) assumes that all strings begin from the same state $q_i \in Q$. Hence it follows:

$$[\![G_{i'}]\!]_{P_i} = [\![G_{i'}\overrightarrow{\otimes}P_i]\!]_{P_i} \tag{58}$$

This completes the proof. ❐

A. Physical Significance of Projected Distribution Invariance

Given a symbolic language theoretic PFSA model for a physical system of interest, one is often concerned with only certain class of possible future evolutions. For example, in the paradigm of deterministic finite state automata (DFSA) [8], the control requirements are expressed in the form of a specification language or a specification automaton. In that setting, it is critical to determine which state of the specification automaton the system is currently visiting. In contrast, for a PFSA, the issue is the probability of certain class of future evolutions. For example, given a large order model of a physical system, it might be necessary to work with a much smaller order PFSA, that has the same long-term behavior with respect to a specified set of event strings. Although projective composition may incur a representation error in general, the long-term distribution over the states of the projected model is preserved as shown in Theorem 5.3. The idea is further clarified in the commutative diagram of Fig. 9.

Probabilistic synchronous composition is an exact representation with no loss of statistical information; but the

$$\begin{tikzcd}[column sep=huge, row sep=huge]
P_i \arrow[r, "\otimes G_{i'}"] \arrow[d, "\vec{\otimes} G_{i'}"'] \arrow[dr, "\llbracket\cdot\rrbracket_{G_{i'}}"] & P_i \otimes G_{i'} \arrow[d, "\llbracket\cdot\rrbracket_{G_{i'}}"] \\
P_i \vec{\otimes} G_{i'} \arrow[r, "\llbracket\cdot\rrbracket_{G_{i'}}"'] & \wp
\end{tikzcd}$$

Fig. 9. Commutative Diagram relating probabilistic composition, projective composition and the original projected distribution

model order increases due to the product automaton construction. On the other hand, the projective composition has the same number of states as the second argument in $(\bullet)\vec{\otimes}(\bullet)$. Both representations have exactly the same projected distribution with respect to a fixed second argument, thus making $\vec{\otimes}$ an extremely useful tool for model order reduction. Algorithm 3 computes the projected composition of two arbitrary PFSA.

Algorithm 3: Computation of Projected Composition

input : $P_i = (Q, \Sigma, \delta, q_i, \tilde{\pi}), G_{i'} = (Q', \Sigma, \delta', q_{i'}, \tilde{\pi}')$
output: $P_i \vec{\otimes} G_{i'}$
begin
  Compute $P_i \otimes G_{i'} = (Q \times Q', \Sigma, \delta^{\otimes}, (q_i, q'_{i'}), \tilde{\pi}^{\otimes})$;
  /*See Definition 4.5*/
  Compute $\wp$; /* State Prob. for $P_i \otimes G_{i'}$ (Def. 2.3) */
  Set up matrix $\mathbf{T}$ s.t. $\mathbf{T}_{jk} = \wp((q_j, q'_k))$;
  Compute $\tilde{\pi}^{\vec{\otimes}} = \mathbf{T}\tilde{\pi}$;
  return $P_i \vec{\otimes} G_{i'} = (Q', \Sigma, \delta', q_{i'}, \tilde{\pi}^{\vec{\otimes}})$;
end

B. Incurred Error in Projective Composition

Given any two PFSA P_i and $G_{i'}$, the incurred error in projective composition operation $P\vec{\otimes}G_{i'}$ is quantified in the pseudo-metric defined in Section 4 as follows:

$$\boldsymbol{\nu}_{\alpha,s}(P_i, P_i \vec{\otimes} G_{i'}) \tag{59}$$

Next we establish a sufficient condition for guaranteeing zero incurred error in projective composition.

Theorem 5.4: *For arbitrary PFSA $P_i = (Q, \Sigma, \delta, q_i, \tilde{\pi})$ and $G_{i'} = (Q', \Sigma, \delta', q'_{i'}, \tilde{\pi}')$ with corresponding probabilistic Nerode equivalence relations $\mathcal{N}$ and $\mathcal{N}'$, we have*

$$\mathcal{N} \leqq \mathcal{N}' \Longrightarrow \boldsymbol{\nu}_{\alpha,s}(G_{i'}, G_{i'} \vec{\otimes} P_i) = 0$$

Proof: $\mathcal{N} \leqq \mathcal{N}'$ implies that there exists a possibly non-injective map $f : Q \to Q'$ such that

$$\forall x \in \Sigma^\star,\ \delta^\star(q_i, x) = q_j \in Q \Longrightarrow \delta^\star(q'_{i'}, x) = f(q_j) \in Q'$$

It then follows from Definition 5.1 that

$$\vartheta(q'_{k'}, q_j) = 0 \text{ if } f(q_j) \neq q'_{k'}$$

Denoting $G_{i'} \otimes P_i = (Q \times Q', \Sigma, \delta^{\otimes}, (q_i, q'_{i'}), \tilde{\pi}^{\otimes})$ and $G_{i'} \vec{\otimes} P_i = (Q, \Sigma, \delta, q_i, \tilde{\pi}^{\vec{\otimes}})$, we have fron Definition 5.1 that

$$\begin{aligned}
\tilde{\pi}^{\vec{\otimes}}(q_j, \sigma) &= \frac{\sum_{q'_{k'} \in Q'} \vartheta(q'_{k'}, q_j) \tilde{\pi}^{\otimes}((q'_{k'}, q_j), \sigma)}{\sum_{q'_{k'} \in Q'} \vartheta(q'_{k'}, q_j)} \\
&= \tilde{\pi}^{\otimes}((f(q_j), q_j), \sigma) = \tilde{\pi}'(f(q_j), \sigma)
\end{aligned}$$

where the last step follows from Definition 4.5. The proof is completed by noting

$$\begin{aligned}
\forall x \in \Sigma^\star,\ \mathbb{H}(G_{i'})(x) &= \tilde{\pi}'(q'_{i'}, x) = \tilde{\pi}^{\otimes}((f(q_i), q_i), x) \\
&= \tilde{\pi}^{\vec{\otimes}}(q_i, x) = \mathbb{H}(G_{i'} \vec{\otimes} P_i)(x)
\end{aligned}$$

❐

Example 5.1: *The results of Section 5 are illustrated considering the PFSA models described in Example 4.1. Given the PFSA models $P^1_{q1} = (\{q_1, q_2, q_3\}, \Sigma, \delta^1, q_1, \tilde{\pi}^1)$ and $P^2_{qA} = (\{q_A, q_B\}, \Sigma, \delta^2, q_A, \tilde{\pi}^2)$ (See Eqns. (40) and (41)), we compute the projected compositions $P^1_{q1} \vec{\otimes} P^2_{qA} = (\{q_A, q_B\}, \Sigma, \delta^2, q_A, \tilde{\pi}^{12})$ and $P^2_{qA} \vec{\otimes} P^1_{q1} = (\{q_1, q_2, q_3\}, \Sigma, \delta^1, q_1, \tilde{\pi}^{21})$. The synchronous compositions $P^1_{q1} \otimes P^2_{qA}$ and $P^2_{qA} \otimes P^1_{q1}$ were computed in Example 4.1 and are shown in Fig. 8. Denoting the associated stochastic transition matrices for $P^1_{q1} \otimes P^2_{qA}$ and $P^2_{qA} \otimes P^1_{q1}$ as Π^{12} and Π^{21} respectively, we note:*

$$\Pi^{12} = \begin{bmatrix} 0.20\,0\,0\,0.8 \\ 0\,0.3.7\,0\,0 \\ .40\,0\,0\,0.60 \\ 0.20\,0\,0\,0.8 \\ 0\,0.3.7\,0\,0 \\ .40\,0\,0\,0.60 \end{bmatrix}, \Pi^{21} = \begin{bmatrix} 0.90\,0\,0\,0.1 \\ 0\,0.9.1\,0\,0 \\ .90\,0\,0\,0.10 \\ 0.70\,0\,0\,0.3 \\ 0\,0.7.3\,0\,0 \\ .70\,0\,0\,0.30 \end{bmatrix} \begin{array}{l} \cdots(q_1, q_A) \\ \cdots(q_2, q_A) \\ \cdots(q_3, q_A) \\ \cdots(q_1, q_B) \\ \cdots(q_2, q_B) \\ \cdots(q_3, q_B) \end{array}$$

The stable probability distributions $\wp^{12}$ and $\wp^{21}$ are computed to be:

$$\wp^{12} = [0.1458\ 0.0695\ 0.0864\ 0.2017\ 0.2186\ 0.2780] \tag{60a}$$

$$\wp^{21} = [0.2917\ 0.2917\ 0.2917\ 0.0417\ 0.0417\ 0.0417] \tag{60b}$$

Using Algorithm 3, we compute the event generating functions $\tilde{\Pi}^{12}$ and $\tilde{\Pi}^{21}$ as:

$$\tilde{\Pi}^{12} = \begin{bmatrix} 0.7197 & 0.2803 \\ 0.6891 & 0.3109 \end{bmatrix}, \tilde{\Pi}^{21} = \begin{bmatrix} 0.1250 & 0.8750 \\ 0.1250 & 0.8750 \\ 0.1250 & 0.8750 \end{bmatrix} \tag{61}$$

We note that the stable distributions for $P^1_{q1} \vec{\otimes} P^2_{qA}$ and $P^2_{qA} \vec{\otimes} P^1_{q1}$ are given by:

$$\overrightarrow{\wp^{12}} = [0.3017\ 0.6983],\ \overrightarrow{\wp^{21}} = [0.3333\ 0.3333\ 0.3333] \tag{62}$$

The operations are illustrated in Figs. 10 and 11 and invariance of the projected distribution is checked as follows:

$$\wp^{12}(1) + \wp^{12}(2) + \wp^{12}(3) = 0.3017 = \overrightarrow{\wp^{12}}(1) \tag{63a}$$

$$\wp^{12}(4) + \wp^{12}(5) + \wp^{12}(6) = 0.6983 = \overrightarrow{\wp^{12}}(2) \tag{63b}$$

$$\wp^{21}(2) + \wp^{21}(5) = 0.333 = \overrightarrow{\wp^{21}}(2) \tag{63c}$$

$$\wp^{21}(3) + \wp^{21}(6) = 0.333 = \overrightarrow{\wp^{21}}(3) \tag{63d}$$

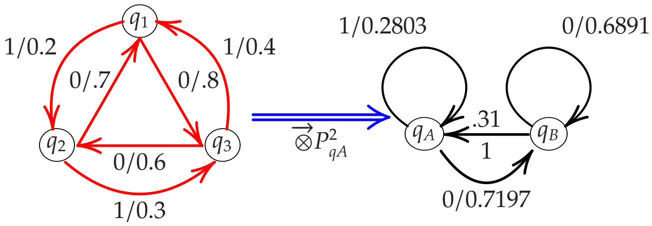


Fig. 10. P^1_{q1} projectively composed with P^2_{qA}

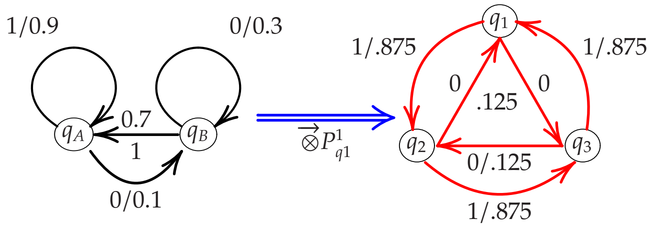


Fig. 11. P^2_{qA} projectively composed with P^1_{q1}

6. An Engineering Application of Pattern Recognition

Projective composition is applied to a symbolic pattern identification problem. Continuous-valued data from a laser ranging array in a sensor fusion test bed are fed to a symbolic model reconstruction algorithm (CSSR) [1] to yield probabilistic finite state models over a four-letter alphabet. A maximum entropy partitioning scheme [3] is employed to create the symbolic alphabet on the continuous time series. Figure 12 depicts the results from four different experimental runs. Two of those runs in the top two rows of Fig. 12 correspond to a human subject moving in the sensor field; the other two runs in the bottom two rows correspond to a robot representing an unmanned ground vehicle (UGV). The symbolic reconstruction algorithm yields PFSA having disparate number of states in each of the above four cases (i.e., two each for the human subject and the robot), with their graph structures being significantly different. The resulting patterns (i.e., state probability vectors) for these PFSA models in each of the four cases are shown on the left side of Fig. 12. The models are then projectively composed with a 64 state D-Markov machine [2] having alphabet size = 4 and depth = 3. The resulting pattern vectors are shown on the right hand column of Fig. 12. The four rows in Fig. 12 demonstrate the applicability of projective composition to statistical pattern classification; the state probability vectors of projected models unambiguously identify the respective patterns of a human subject and an UGV.

7. Summary, Conclusions & Future Work

This paper presents a rigorous measure-theoretic approach to probabilistic finite state machines. Key concepts from classical language theory such as the Nerode equivalence relation is generalized to the probabilistic paradigm and the existence and uniqueness of minimal representations for PFSA is established. Two binary operations, namely, probabilistic synchronous composition and projective composition of PFSA are introduced and their properties are investigated. Numerical examples have been provided for clarity of exposition. The applicability of the defined binary operators has been demonstrated on experimental data from a laboratory test bed in a pattern identification and classification problem. This paper lays the framework for three major directions for future research and the associated applications.

- **Probabilistic Non-regular Languages:** Since projective composition can be used to obtain smaller order models with quantifiable error, the possibility of projectively composing infinite state probabilistic models with finite state machines must be investigated. The extension of the theory developed in this paper to non-regular probabilistic languages would prove invaluable in handling strictly non-Markovian models in the symbolic paradigm, especially physical processes that fail to have the semi-Martingle property, *e.g.*, fractional Brownian motion [19]. Future work will investigate language-theoretic non-regularity as the symbolic analogue to chaotic behavior in the continuous domain.
- **Optimal Control:** The reported measure-theoretic approach to optimal supervisor design in PFSA models will be extended in the light of the developments reported in this paper to situations where the control specification is given as weights on the states of DFSA models disparate from the plant under consideration. Such a generalization would allow the fusion of Ramadge and Wonham's constraint based supervision approach [8] with the measure-theoretic approach reported in [10][11]. This new control synthesis tool would prove invaluable in the design of event driven controllers in probabilistic robotics.
- **Pattern Identification:** Preliminary application in pattern classification has already been demonstrated in Section 6. Future research will formalize the approach and investigate methodologies for optimally choosing the plant model on which to project the constructed PFSA to yield maximum algorithmic performance. Future investigations will explore applicability of the structural transformations developed in this paper for the fusion, refinement and computation of bounded order symbolic

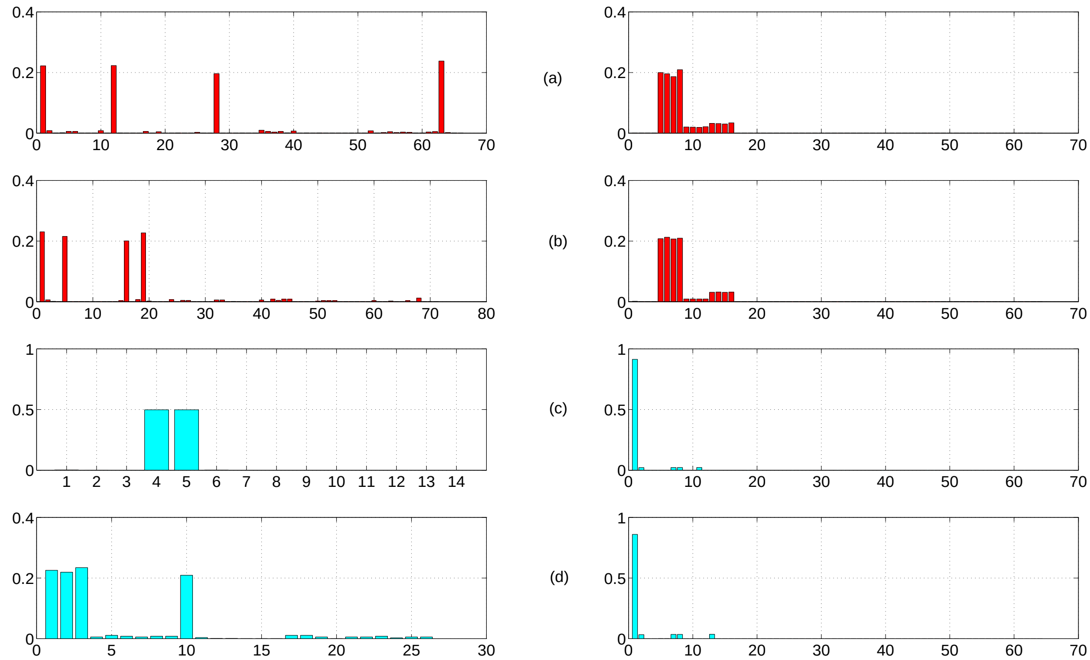


Fig. 12. Experimental Validation of Projective Composition in Pattern Recognition: (a) and (b) correspond to ranging data for a human subject in sensor field; (c) and (d) correspond to an UGV

models of observed system behavior in complex dynamical systems.

ACKNOWLEDGEMENTS

The authors would like to thank Dr. Eric Keller for his valuable contribution in obtaining the experimental results.

REFERENCES

[1] C. R. Shalizi and K. L. Shalizi, "Blind construction of optimal nonlinear recursive predictors for discrete sequences," in *AUAI '04: Proceedings of the 20th conference on Uncertainty in artificial intelligence*, (Arlington, Virginia, United States), pp. 504–511, AUAI Press, 2004.

[2] A. Ray, "Symbolic dynamic analysis of complex systems for anomaly detection," *Signal Processing*, vol. 84, no. 7, pp. 1115–1130, 2004.

[3] V. Rajagopalan and A.Ray, "Symbolic time series analysis via wavelet-based partitioning," *Signal Processing*, vol. 86, no. 11, pp. 3309–3320, 2006.

[4] V. Garg, "An algebraic approach to modeling probabilistic discrete event systems," *Proceedings of 1992 IEEE Conference on Decision and Control*, pp. 2348–2353, Tucson, AZ, December 1992.

[5] V. Garg, "Probabilistic lnaguages for modeling of DEDs," *Proceedings of 1992 IEEE Conference on Information and Sciences*, pp. 198–203, Princeton, NJ, March 1992.

[6] M. Lawford and W. Wonham, "Supervisory control of probabilistic discrete event systems," *Proceedings of 36th Midwest Symposium on Circuits and Systems*, pp. 327–331, 1993.

[7] R. Kumar and V. Garg, "Control of stochastic discrete event systems modeled by probabilistic languages," *IEEE Transactions on Automatic Control*, vol. 46, no. 2, pp. 593–606, 2001.

[8] P. J. Ramadge and W. M. Wonham, "Supervisory control of a class of discrete event processes," *SIAM J. Control and Optimization*, vol. 25, no. 1, pp. 206–230, 1987.

[9] A. Ray, "Signed real measure of regular languages for discrete-event supervisory control," *Int. J. Control*, vol. 78, no. 12, pp. 949–967, 2005.

[10] I. Chattopadhyay, "Quantitative control of probabilistic discrete event systems," *PhD Dissertation, Dept. of Mech. Engg. Pennsylvania State University, http:// etda.libraries.psu.edu / theses / approved / WorldWideIndex / ETD-1443*, 2006.

[11] I. Chattopadhyay and A. Ray, "Language-measure-theoretic optimal control of probabilistic finite-state systems," *Int. J. Control*, vol. 80, no. 8, pp. 1271–1290, 2007.

[12] I. Chattopadhyay and A. Ray, "Renormalized measure of regular languages," *Int. J. Control*, vol. 79, no. 9, pp. 1107–1117, 2006.

[13] J. E. Hopcroft, R. Motwani, and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation, 2nd ed.* Addison-Wesley, Boston, MA, USA, 2001.

[14] W. J. Harrod and R. J. Plemmons, "Comparison of some direct methods for computing the stationary distributions of markov chains," *SIAM J. Sci. Statist. Comput.*, vol. 5, pp. 453–469, 1984.

[15] J. G. Kemeny and J. L. Snell, *Finite Markov Chains*. New York: Springer, 2 ed., 1960.

[16] W. Stewart, *Computational Probability: Numerical methods for computing stationary distribution of finite irreducible Markov chains*. New York: Springer, 1999.

[17] W. Rudin, *Real and Complex Analysis, 3rd ed.* McGraw Hill,

New York, 1988.
[18] R. Bapat and T. Raghavan, *Nonnegative matrices and Applications*. Cambridge University Press, 1997.
[19] L. Decreusefond and A. Ustunel, "Stochastic analysis of the fractional brownian motion," *Potential Analysis*, vol. 10, no. 2, pp. 177–214, 1999.

International Journal of Control
Vol. 80, No. 8, August 2007, 1271–1290

Taylor & Francis
Taylor & Francis Group

Language-measure-theoretic optimal control of probabilistic finite-state systems

I. CHATTOPADHYAY and A. RAY*

Mechanical Engineering Department, The Pennsylvania State University, University Park, PA 16802, USA

(*Received 10 October 2006; in final form 17 February 2007*)

Supervisory control theory for discrete event systems, introduced by Ramadge and Wonham, is based on a non-probabilistic formal language framework. However, models for physical processes inherently involve modelling errors and noise-corrupted observations, implying that any practical finite-state approximation would require consideration of event occurrence probabilities. Building on the concept of signed real measure of regular languages, this paper formulates a comprehensive theory for optimal control of finite-state probabilistic processes. It is shown that the resulting discrete-event supervisor is optimal in the sense of elementwise maximizing the renormalized langauge measure vector for the controlled plant behaviour and is efficiently computable. The theoretical results are validated through several examples including the simulation of an engineering problem.

1. Introduction

Supervisory control theory (SCT) of discrete-event systems (DES), pioneered by Ramadge and Wonham (1987), models a physical or human-engineered process as a finite-state language generator and constructs a supervisor that attempts to constrain the "supervized" plant behaviour within a specification language. The original theory is based on a deterministic language framework. Although allowing non-determinism in the sense that more than one continuation of a generated event trace (i.e., a string) is possible, no attempt is made to quantify this randomness. As Wonham himself observes in Lawford and Wonham (1993), "the choice of a possible continuation of a string is made by some internal structure unmodeled by the systems designer". The notion of probabilistic languages in the context of studying qualitative stochastic behaviour of discrete-event systems first appears in Garg (1992a, b), where the concept of p-languages ('p' implying probabilistic) is introduced and an algebra is developed to model probabilistic languages based on concurrency (Milner 1989). A regular p-language is essentially a set of prefix-closed traces of events, generated by a finite-state automaton with probabilities associated with the transitions. A p-language-theoretic model differs in several important aspects from other discrete-event models of stochastic analysis such as Markov chains (Cassandras and Lafortune 1999), stochastic Petri nets (Molloy 1982, Chung *et al.* 1994), probabilistic automata (Rabin 1963, Paz 1971, Doberkat 1981), and fuzzy models (Lee and Zadeh 1969). Garg *et al.* (1999) and Kumar and Garg (2001) provide a brief comparison of the p-language-theoretic modelling paradigm with the above-mentioned theories.

Lawford and Wonham (1993) have attempted to extend discrete-event (SCT) to plants modelled by p-languages, where a formal statement of the probabilistic supervisory control problem (PSCP) first appears and the notion of probabilistic supervision is introduced by random disabling of controllable events. The key difference from other stochastic supervision approaches (e.g., Mortzavian 1993) lies in the fact that the computed probabilistic supervisor is not allowed to change the

*Corresponding author. Email: axr2@psu.edu

International Journal of Control
ISSN 0020–7179 print/ISSN 1366–5820 online © 2007 Taylor & Francis
http://www.tandf.co.uk/journals
DOI: 10.1080/00207170701286322

underlying plant dynamics in the following sense: "The probabilistic effect of random disablement is determined entirely by the plant". The control objective is specified as a p-language and necessary and sufficient conditions are derived for existence of a probabilistic supervisor that attempts to restrict the plant language within the control specification in a probabilistic sense. The theory of supervision of p-languages is further developed by Kumar and Garg (2001), where the control objective is specified as upper and lower bound constraints. The upper bound is a non-probabilistic language that serves as a legality constraint, while the lower bound is a p-language. This relatively relaxed approach to control objective specification allows for a non-probabilistic supervisor that attempts to cut down illegal event traces, while ensuring that legal traces occur with probabilities greater than or equal to that specified by the lower bound. Intuitively, the designed supervisor stops "bad" strings from occurring while guaranteeing that "good" strings occur with some minimum pre-set frequency. However, construction of such a control objective specification may not be possible in many applications (e.g., battlefield command, control, communications, and intelligence (C^3I) (Phoha *et al.* 2002)), especially if the decisions are to made in real time. For the theory to be useful in practice, one must generate the specification from the definition of the physical problem at hand. Given that one has to come up with a non-probabilistic language to serve as the upper legality constraint and a probabilistic language to serve as the lower bound, this goal may not always be achievable. The situation becomes worse for non-stationary stochastic environments, where the control specifications may have to be updated online.

A significantly simplified approach to the above problem is reported by Ray (2005) and Ray *et al.* (2005), where the control objective is specified as characteristic weights on the states of the plant automaton. These weights are normalized in the interval $[-1, 1]$ with positive weights assigned to good states and negative weights to bad states. A signed real measure of regular languages (of event traces) is defined as a function of the characteristic weights and the state transition probabilities; and supervisory control laws are synthesized by elementwise maximizing the language measure vector (Ray *et al.* 2004, 2005). Intuitively, the supervisor ensures that the generated event traces cause the plant to visit the "good" states while attempting to avoid the "bad" states in a probabilistic sense. As mentioned earlier, Kumar and Garg's work on supervisory control of probabilistic automata (Kumar and Garg 2001) also has a notion of "good" and "bad" strings. However, the classification is strictly binary; the theory has no way of saying if one "good" string is "better" than another "good" string and *vice versa*. This implies that the supervisor must eliminate all bad strings and hence may not be optimal, or fail to exist if the conditions defined in Kumar and Garg (2001) are not satisfied. In contrast, in the measure-theoretic approach (Ray 2005), the event traces are less or more desirable in a continuous scale with the supervisor optimizing the controlled plant behaviour to ensure that the "most" desirable strings occur "most" often. This has an immediate advantage that the problem of existence disappears; the optimal supervisor always exists and can be computed effectively with polynomial complexity. The latter approach is, in one sense, closer to Markov chain modelling since the control specification is state-based. However, as shown by Ray (2005), this does not restrict the modeling power of the technique. It is shown in Kumar and Garg (2001) that, in general, maximally permissive supervisors are non-unique. For the measure-theoretic approach, however, the optimization is shown to yield unique maximal permissiveness among all optimal supervisors (Ray *et al.* 2004, 2005).

Optimal control in the context of discrete event dynamic systems has been addressed earlier by several investigators as cited in (Ray *et al.* 2004). For example, Sengupta and Lafortune (1998) have analysed non-probabilistic DES with assigned event and control costs; the optimal supervisor is computed in the framework of dynamic programming (DP) with two critical assumptions to guarantee polynomial complexity of the solution: all costs are strictly positive and there is only one marked state (Sengupta and Lafortune 1998, p. 34). The work reported in Ray (2005) and Ray *et al.* (2005) is different in the sense that the latter deals with probabilistic automata and the optimization, even in the completely general case, has guaranteed polynomial complexity of $O(n^3)$, where n is the number of states in the unsupervised plant model. The measure-theoretic approach was originally reported for a restricted class of terminating p-languages Ray (2005) and Ray *et al.* (2005); and this restriction has been eliminated in a subsequent publication (Chattopadhyay and Ray 2006a).

The notion of terminating and non-terminating automata is originally due to Garg (19992a, b). A probabilistic automaton is terminating if there exist states at which the sum of the probabilities of all defined events is strictly less than 1. The interpretation is that the difference of the sum from 1 is the probability that the plant terminates operation at that particular state. It is shown in Ray (2005) that the language measure vector can be expressed as $[\mathbb{I} - \boldsymbol{\Pi}]^{-1}\boldsymbol{\chi}$ where $\boldsymbol{\Pi}$ is the transition probability matrix and $\boldsymbol{\chi}$ is the characteristic vector, where π_{ij} is the probability of transition from the ith state to the jth state and χ_i is the characteristic weight of the

state i). A sufficient condition for the inverse of $[\mathbb{I} - \Pi]$ to exist is that $\sum_j \pi_{ij} < 1 \ \in i$, i.e., the plant has a strictly non-zero probability of termination from each state.

This paper eliminates the above restrictive assumption by adopting the recently reported renormalized measure of regular languages (Chattopadhyay and Ray 2006a) as the performance index. It also extends the measure-theoretic concept for optimal control of terminating plants (Ray *et al.* 2004) to non-terminating plant models, which requires a minor modification of the control philosophy as explained below.

Supervisors in the SCT paradigm are allowed to affect the underlying plant behaviour by selectively disabling controllable events (Ramadge and Wonham 1987). In case of terminating p-languages, a similar approach suffices; the supervisor selectively nulls the occurrence probability of controllable events to achieve the desired control objective. However, the non-terminating case poses a problem since any such disabling action converts the system to a terminating p-language (i.e., the probabilities of the events defined at a state fail to add up to 1). The solution (Kumar and Garg 2001) is to proportionately increase the probabilities of the remaining enabled events at the state at which event disabling is undertaken. An alternative approach is proposed in this paper, where each disabled event creates a self loop at the state (at which the event was generated) with occurrence probability of the original transition.

The paper is organized in six sections and an appendix. Section 2 lays down the basic framework of the analysis and briefly reviews the original notion of language measure (Ray 2005) and its renormalization (Chattopadhyay and Ray 2006a). Section 3 formulates the optimal control problem based on the concept of renormalized measure and presents the key results. Section 4 presents a solution of the optimal control problem and derives the necessary algorithms for its implementation. Section 5 presents an engineering example, where the optimal supervisor is designed for a three-processor message decoding system. The paper is summarized and concluded in § 6 along with recommendations for future research. Appendix G establishes bounds on the derivatives of the renormalized measure that is necessary for formulation of the optimal control law in § 3.

2. Preliminary concepts

This section briefly reviews the concept of signed real measure of regular languages Ray (2005) and Ray *et al.* (2005) followed by a review of the notion of renormalized measure and the pertinent notations used in the sequel.

2.1 *Brief review of language measure*

Let the plant behaviour be modelled as a deterministic finite state automaton (DFSA) as $G_i \triangleq (Q, \Sigma, \delta, q_i, Q_m)$, where Q is the finite set of states with $|Q| = n$, and $q_i \in Q$ is the initial state; Σ is the (finite) alphabet of events with $|\Sigma| = m$; the Kleene closure of Σ is denoted as $\Sigma^\star$ that is the set of all finite-length strings of events including the empty string ε; the (possibly partial) function $\delta: Q \times \Sigma \to Q$ represents state transitions and $\delta^\star: Q \times \Sigma^\star \to Q$ is an extension of δ; and $Q_m \subseteq Q$ is the set of marked (i.e., accepted) states.

Definition 1: The language $L(G_i)$ generated by the *DFSA* G_i is defined as $L(G_i) = \{s \in \Sigma^* \mid \delta^*(q_i, s) \in Q\}$.

Definition 2: The marked language $L_m(G_i)$ by the *DFSA* G_i is defined as $L_m(G_i) = \{s \in \Sigma^* \mid \delta^*(q_i, s) \in Q_m\}$.

The language $L(G_i)$ of the DFSA G_i is partitioned as the non-marked and the marked languages, $L^o(G_i) \triangleq L(G_i) - L_m(G_i)$ and $L_m(G_i)$, consisting of event strings that, starting from $q_i \in Q$, terminate at one of the non-marked states in $Q - Q_m$ and one of the marked states in Q_m, respectively. The set Q_m is partitioned into Q_m^+ and Q_m^- where Q_m^+ contains all good marked states that one may desire to terminate on, and Q_m^- contains all bad marked states that one would attempt to avoid terminating on, although it may not always be possible to bypass a bad state before reaching a good state. The marked language $L_m(G_i)$ is further partitioned into $L_m^+(G_i)$ and $L_m^-(G_i)$ consisting of good and bad strings that, starting from q_i, terminate on Q_m^+ and Q_m^-, respectively.

A signed real measure $\mu: 2^{\Sigma^\star} \to R \triangleq (-\infty, \infty)$ is constructed for quantitative evaluation of every event string $s \in \Sigma^\star$. The language $L(G_i)$ is decomposed into null, i.e., $L^o(G_i)$, positive, i.e., $L_m^+(G_i)$, and negative, i.e., $L_m^-(G_i)$ sublanguages.

Definition 3: The language of all strings that, starting at a state $q_i \in Q$, terminates on a state $q_j \in Q$, is denoted as $L(q_i, q_j)$. That is,

$$L(q_i, q_j) \triangleq \{s \in L(G_i) : \delta^\star(q_i, s) = q_j\}. \tag{1}$$

Definition 4: The characteristic function that assigns a signed real weight to each state $q_i, i = 1, 2, \ldots, n$, is defined as: $\chi: Q \to [-1, \ 1]$ such that

$$\chi(q_j) \in \begin{cases} [-1, 0) & \text{if } q_j \in Q_m^- \\ \{0\} & \text{if } q_j \notin Q_m \\ (0, 1] & \text{if } q_j \in Q_m^+ \end{cases}$$

Definition 5: The event cost is conditioned on a *DFSA* state at which the event is generated, and is defined as $\tilde{\pi} : \Sigma^\star \times Q \to [0, 1]$ such that $\forall q_j \in Q, \forall \sigma_k \in \Sigma$, $\forall s \in \Sigma^\star$,

1. $\tilde{\pi}[\sigma_k, q_j] \triangleq \tilde{\pi}_{jk} \in [0, 1)$; $\sum_k \tilde{\pi}_{jk} < 1$;
2. $\tilde{\pi}[\sigma, q_j] = 0$ if $\delta(q_j, \sigma)$ is undefined; $\tilde{\pi}[\epsilon, q_j] = 1$;
3. $\tilde{\pi}[\sigma_k s, q_j] = \tilde{\pi}[\sigma_k, q_j]\ \tilde{\pi}[s, \delta(q_j, \sigma_k)]$.

The event cost matrix, denoted as $\widetilde{\boldsymbol{\Pi}}$-matrix, is defined as

$$\widetilde{\Pi} = \begin{bmatrix} \tilde{\pi}_{11} & \dots & \tilde{\pi}_{1m} \\ \vdots & \ddots & \vdots \\ \tilde{\pi}_{n1} & \dots & \tilde{\pi}_{nm} \end{bmatrix}$$

An application of the induction principle to part (3) in Definition 5 shows $\tilde{\pi}[st, q_j] = \tilde{\pi}[s, q_j]\tilde{\pi}[t, \delta^*(q_j, s)]$. The condition $\Sigma_k \tilde{\pi}_{jk} < 1$ provides a sufficient condition for the existence of the real signed measure as discussed in Ray (2005) along with additional comments on the physical interpretation of the event cost.

Now let us define the measure of a sublanguage of the plant language $L(G_i)$ in terms of the signed characteristic function χ and the non-negative event cost $\tilde{\pi}$.

Definition 6: The signed real measure μ of a singleton string set $\{s\} \subseteq L(q_i, q_j) \subseteq L(G_i) \in 2^{\Sigma^\star}$ is defined as

$$\mu(\{s\}) \triangleq \tilde{\pi}(s,\, q_i)\chi(q_j) \quad \forall s \in L(q_i, q_j).$$

The signed real measure of $L(q_i, q_j)$ is defined as

$$\mu\big(L(q_i, q_j)\big) \triangleq \sum_{s \in L(q_i,\, q_j)} \mu(\{s\})$$

and the signed real measure of a *DFSA* G_i, initialized at the state $q_i \in Q$, is denoted as

$$\mu_i \triangleq \mu(L(G_i)) = \sum\nolimits_j \mu\big(L(q_i, q_j)\big).$$

Definition 7: The state transition cost of the *DFSA* is defined as a function π: $Q \times Q \to [0, 1)$ such that

$$\pi(q_j, q_k) = \begin{cases} 0 & \text{if } \{\sigma \in \Sigma\colon \delta(q_j, \sigma) = q_k\} = \emptyset \\ \displaystyle\sum_{\sigma \in \Sigma: \delta(q_j, \sigma) = q_k} \tilde{\pi}(\sigma, q_j) \triangleq \pi_{jk} & \text{otherwise} \end{cases}. \tag{2}$$

The state transition cost matrix, denoted as $\boldsymbol{\Pi}$-matrix, is defined as

$$\boldsymbol{\Pi} = \begin{bmatrix} \pi_{11} & \dots & \pi_{1n} \\ \vdots & \ddots & \vdots \\ \pi_{n1} & \dots & \pi_{nn} \end{bmatrix}.$$

It has been shown in (Ray 2005 and Ray *et al.* 2005) that the measure $\mu_i \triangleq \mu(L(G_i))$ of the language $L(G_i)$, with the initial state q_i, can be expressed as $\mu_i = \sum_j \pi_{ij}\, \mu_j + \chi_i$ where $\chi_i \triangleq \chi(q_i)$. Equivalently, in vector notation

$$\boldsymbol{\mu} = \boldsymbol{\Pi}\boldsymbol{\mu} + \boldsymbol{\chi} \Longrightarrow \boldsymbol{\mu} = [\mathbb{I} - \boldsymbol{\Pi}]^{-1}\boldsymbol{\chi}, \tag{3}$$

where the measure vector $\boldsymbol{\mu} \triangleq [\mu_1\ \mu_2\ \cdots\ \mu_n]^T$ and the characteristic vector $\boldsymbol{\chi} \triangleq [\chi_1\ \chi_2\ \cdots\ \chi_n]^T$; and the condition $\sum_j \tilde{\pi}_{ij} < 1\ \in i$ (see Definition 5) is sufficient for the inverse to exist.

Although the preceding analysis reported in (Ray 2005 and Ray *et al.* 2005) was intended for non-probabilistic regular languages, the event costs can be easily interpreted as occurrence probabilities. As such the $\widetilde{\boldsymbol{\Pi}}$-matrix is analogous to the morph matrix of a Markov chain in the sense that an element $\tilde{\pi}_{ij}$ represents the probability of the jth event occurring at the ith state with the exception that the strict inequality condition $\sum_j \tilde{\pi}_{ij} < 1$ is enforced instead of satisfying the equality. Equivalently, the $\boldsymbol{\Pi}$-matrix is analogous to the state transition probability matrix of a Markov chain in the sense that an element π_{jk} is analogous to the transition probability from state q_j to state q_k with the exception that the strict inequality condition $\sum_k \pi_{jk} < 1$ is enforced instead of satisfying the equality. It follows that the preceding analysis is applicable to the case of terminating probabilistic languages (Garg *et al.* 1992a, b) that have a strictly non-zero probability of termination at each state.

Let Σ^u denote the set of all unmodelled events in the terminating plant. A new unmarked absorbing state q_{n+1}, called the dump state (Ramadge and Wonham 1987), is created and the transition function δ is extended to $\delta_{\text{ext}} : (Q \cup \{q_{n+1}\}) \times (\Sigma \cup \Sigma^u) \to (Q \cup \{q_{n+1}\})$. The residue $\theta_j = 1 - \sum_k \pi_{jk}$ denotes the probability of transition from q_j to q_{n+1}. Consequently, the $\boldsymbol{\Pi}$-matrix (see Definition 7) is augmented to obtain the stochastic state transition probability matrix as

$$\boldsymbol{\Pi}_{\text{aug}} = \begin{bmatrix} \pi_{11} & \pi_{12} & \dots & \pi_{1n} & \theta_1 \\ \pi_{21} & \pi_{22} & \dots & \pi_{2n} & \theta_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \pi_{n1} & \pi_{n2} & \dots & \pi_{nn} & \theta_n \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}. \tag{4}$$

Since the dump state q_{n+1} is not marked (Ramadge and Wonham 1987), it follows from Definition 4 that the corresponding state weight $\chi_{n+1} = 0$. Hence, the χ-vector is augmented as

$$\boldsymbol{\chi}_{\text{aug}} = [\boldsymbol{\chi}^T\ 0]^T. \tag{5}$$

Denoting $\boldsymbol{\Theta} = [\theta_1\ \theta_2 \cdots \theta_n]^T$, where $\theta_i \in (0, 1)$ is the probability of transition from the state q_i to the dump state, it follows from equations (4) and (5) that the measure of the augmented system (Chattopadhyay and Ray 2006a) is

$$\boldsymbol{\mu}_{\text{aug}}(\boldsymbol{\Theta}) = [\boldsymbol{\mu}(\boldsymbol{\Theta})^T\ 0]^T. \tag{6}$$

Then, the event cost matrix (see Definition 5) and the state transition cost matrix (see Definition 7) can be represented as

$$\widetilde{\boldsymbol{\Pi}}(\boldsymbol{\Theta}) = \Big[\mathbb{I} - \text{Diag}[\boldsymbol{\Theta}]\Big]\widetilde{\mathbf{P}} \quad \text{and} \quad \boldsymbol{\Pi}(\boldsymbol{\Theta}) = \Big[\mathbb{I} - \text{Diag}[\boldsymbol{\Theta}]\Big]\mathbf{P}, \tag{7}$$

where $\widetilde{\mathbf{P}}$ and $\mathbf{P}$ are both stochastic matrices (Bapat and Raghavan 1997), i.e., $\Sigma_j \widetilde{\mathbf{P}}_{ij} = 1\ \forall i \in \{1, \ldots, m\}$ and $\Sigma_j \mathbf{P}_{ij} = 1\ \forall i \in \{1, \ldots, n\}$.

If the probability of termination (or equivalently the probability of transition to the dump state) is equal for all states, $q_i \in Q$, i.e., $\theta_i = \theta\ \ \forall i \in \{1, 2, \ldots, n\}$, then equation (6) is expressed as

$$\boldsymbol{\mu}_{\text{aug}}(\theta) = [\boldsymbol{\mu}(\theta)^T\ 0]^T \tag{8}$$

Consequently, $\widetilde{\boldsymbol{\Pi}}$ and $\boldsymbol{\Pi}$ in equation (7) are represented as

$$\widetilde{\boldsymbol{\Pi}}(\theta) = (1-\theta)\widetilde{\mathbf{P}} \quad \text{and} \quad \boldsymbol{\Pi}(\theta) = (1-\theta)\mathbf{P} \tag{9}$$

where θ is the uniform probability of termination at all states; and both $\widetilde{\mathbf{P}}$ and $\mathbf{P}$ retain the properties of stochastic matrices (Bapat and Raghavan 1997).

2.2 *Renormalization of language measure*

The notion of language measure has been recently extended to non-terminating models by assuming a uniform non-zero probability of termination (θ) at each state, renormalizing the language measure vector with respect to the probability of termination and computing the limit of the renormalized measure (Chattopadhyay and Ray 2006a) as $\theta \to 0^+$. As the probability of termination approaches zero (i.e., $\theta \to 0^+$), and the plant coincides with the desired non-terminating model in the limit. The construction of renormalized measure is briefly outlined below.

The regular language generated by the *DFSA* under consideration is a sublanguage of the Kleene closure Σ^* of the alphabet Σ, for which the automaton states can be merged into a single state. In that case, the state transition cost matrix $\boldsymbol{\Pi}(\theta)$ degenerates to the 1×1 matrix $[1-\theta]$ and the normalized state weight vector $\boldsymbol{\chi}$ becomes one-dimensional and can be assigned as $\boldsymbol{\chi} = 1$; consequently, the measure vector $\boldsymbol{\mu}(\theta)$ degenerates to the scalar measure θ^{-1}. To alleviate the singularity of the matrix operator $[I - \boldsymbol{\Pi}(\theta)]$ at $\theta = 0$, the measure vector $\boldsymbol{\mu}(\theta)$ in (3) is normalized with respect to θ^{-1} to obtain the renormalized measure vector.

Definition 8: The renormalized measure is defined as

$$\boldsymbol{\nu}(\theta) = \theta\ \boldsymbol{\mu}(\theta) = \theta\ [\mathbb{I} - \boldsymbol{\Pi}(\theta)]^{-1}\boldsymbol{\chi} \tag{10}$$

and it follows from (8) that

$$\theta\ \boldsymbol{\mu}_{\text{aug}}(\theta) = [\boldsymbol{\nu}(\theta)^T\ 0]^T. \tag{11}$$

3. Optimal control problem: formulation

The following notations are needed for elementwise comparison of finite-dimensional vectors and matrices for the analysis developed in the sequel.

Notation 1: Let V^a and V^b be $(m \times n)$ real matrices. The following elementwise equality and inequalities imply that

$$\begin{aligned}
\left(V^a \equiv_{\mathbf{E}} V^b\right) &\Leftrightarrow \left(V^a_{ij} = V^b_{ij}\right) \forall i \in \{1, \ldots, n\}\ \forall j \in \{1, \ldots, m\} \\
\left(V^a \neq_{\mathbf{E}} V^b\right) &\Leftrightarrow \left(V^a_{ij} \neq V^b_{ij}\right) \exists\, i \in \{1, \ldots, n\},\ j \in \{1, \ldots, m\} \\
\left(V^a \geqq_{\mathbf{E}} V^b\right) &\Leftrightarrow \left(V^a_{ij} \geq V^b_{ij}\right) \forall i \in \{1, \ldots, n\}\ \forall j \in \{1, \ldots, m\} \\
\left(V^a >_{\mathbf{E}} V^b\right) &\Leftrightarrow \left(V^a_{ij} > V^b_{ij}\right) \forall i \in \{1, \ldots, n\}\ \forall j \in \{1, \ldots, m\}.
\end{aligned}$$

For the terminating plant, investigated in (Ray 2005 and Ray *et al.* 2005), the optimal supervisor selectively disables controllable transitions by setting their occurrence probabilities to zero. This implies that if $\boldsymbol{\Pi}^\star$ and $\boldsymbol{\Pi}$ are the transition probability matrices for the optimally supervised plant and the unsupervised plant, respectively, then

$$\boldsymbol{\Pi}^\star \leqq_{\mathbf{E}} \boldsymbol{\Pi}, \quad \text{i.e., } \pi^\star_{ij} \leq \pi_{ij}.$$

Since for any non-trivial supervisor, there is at least one disabled transition in the supervised plant, i.e.,

$$\exists i, j \text{ such that } \pi_{i,j} > 0 \quad \text{and} \quad \pi^\star_{i,j} = 0$$

it follows that if the unsupervised plant is non-terminating, then any non-trivial supervision will result in a terminating model. The policy of Kumar and

Garg 2001 maintains the non-termination property by proportionately increasing the probabilities of the remaining enabled events at the state at which the disabling action is applied. The first issue here is that the supervisor must be able to affect the event occurrence probabilities, which is more than just inhibiting a transition. The second issue is that there is a possibility of disabling all events defined at a given state if all these events are controllable. In that case, the row sum cannot be maintained at 1 as it becomes strictly equal to zero. Thus, it is necessary to impose special constraints on the unsupervised plant to circumvent this situation. This paper investigates an alternative approach as described below.

Definition 9 (control philosophy)**:** Disabling any transition σ at a given state q results in reconfiguration of the automaton structure as: Set the self-loop $\delta(q,\sigma)=q$ with the occurrence probability of σ from the state q remaining unchanged in the supervised and unsupervised plants.

This is equivalent to adding a self-loop to the state at which the event is being disabled, with the same occurrence probability as the disabled transition.

Preposition 1: For the control philosophy in Definition 9, a supervised plant is non-terminating if and only if the unsupervised plant is non-terminating.

Proof: The proof follows from two lemmas.

Lemma 1: Each row sum of the $\mathbf{\Pi}$-matrix remains unchanged after supervisory actions for the control philosophy in Definition 9.

Proof: Let $\mathbf{\Pi}$ and $\mathbf{\Pi}^\dagger$ be the transition probability matrices for the unsupervised and supervised plants, respectively. Let there be exactly one disabled transition, in which a (controllable) event σ at state q_i is disabled and let the occurrence probability of σ at state q_i be $\tilde{\pi}$. If $\delta(q_i,\sigma)=q_k$, then it follows that

$$\mathbf{\Pi}^\dagger = \mathbf{\Pi} + \begin{array}{cc} \begin{array}{c} k\text{th column} \\ \downarrow \end{array} & \\ \begin{bmatrix} 0 & 0 & \cdots & \cdots & 0 & 0 \\ \vdots & \ddots & & \vdots & \vdots & \vdots \\ 0 & \cdots & \tilde{\pi} & \cdots & -\tilde{\pi} & 0 \\ \vdots & \cdots & 0 & \ddots & 0 & \vdots \\ \vdots & \vdots & \vdots & & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 \end{bmatrix} & \begin{array}{c} \\ \\ \leftarrow i\text{th row} \\ \\ \\ \\ \end{array} \end{array}$$

implying $\sum_j \pi_{ij}^\dagger = \sum_j \pi_{ij}\ \forall i$. The proof follows by induction on the number of disabled events. □

Lemma 2: Self-loops cannot be disabled.

Proof: For the control philosophy in Definition 9, disabling a self-loop at any given state causes regeneration of the self-loop at the same state with identical occurrence probability. □

It is evident from the above two lemmas that each row sum of the reconfigured $\mathbf{\Pi}$-matrix remains invariant. The proof of Proposition 1 is thus complete. □

Remark 1: The control philosophy in Definition 9 is natural in the following sense. If $q_i \rightarrow_\sigma q_k$, and the controllable event σ is disabled at state q_i, then the sole effect of the supervisory action is to prevent the plant from making a transition to the state q_k. That is, the plant is forced to stay at the original state q_i and this is represented by the additional self-loop at state q_i instead of the original arc from q_i to q_k.

The notion of controllability is now clarified in the context of the present paper.

Definition 10 (controllable transitions)**:** For a given plant, transitions that can be disabled in the sense of Definition 9 are defined to be controllable transitions in the sequel.

In view of Definition 10, controllability becomes state-based, i.e., transitions labelled by the same event may be controllable from one state and uncontrollable from some other state. This implies that the event alphabet Σ cannot be partitioned into uncontrollable and controllable events sets as proposed in Ramadge and Wonham (1987). Thus, a controllable transition $q_i \overset{\sigma}{\rightarrow} q_k$ refers to a triple $\{q_i,\sigma,q_k\}$ and the set of all such transitions is denoted by $\mathscr{C}$.

3.1 *Model specification*

Plant models considered in this paper are deterministic finite state automata (DFSA) with well-defined event occurrence probabilities. In other words, the occurrence of events is probabilistic, but the state at which the plant ends up, given a particular event has occurred, is deterministic. Furthermore, no emphasis is laid on the initial state of the plant and it is assumed that the plant may start from any state. Furthermore, having defined the characteristic state weight vector $\boldsymbol{\chi}$, it may not be necessary to specify the set of marked states, because if $\chi_i=0$, then q_i is not marked and if $\chi_i \neq 0$, then q_i is marked. Therefore, plant models with an arbitrary uniform termination probability $\theta \in (0,1)$, i.e., $\theta_i=\theta\ \ \forall i \in \{1,2,\ldots,n\}$, can be completely specified by a sextuple as

$$G(\theta) = \big(Q,\Sigma,\delta,\widetilde{\mathbf{\Pi}}(\theta),\boldsymbol{\chi},\mathscr{C}\big), \tag{12}$$

where $\widetilde{\Pi}(\theta)_{ij}$ is the occurrence probability of event σ_j from state q_i and $\sum_j \widetilde{\Pi}(\theta)_{ij} = 1 - \theta \ \forall i$. An application of (7) with uniform uniform termination probability θ yields an alternative representation of the sextuple in (12).

$$G(\theta) = \big(Q, \Sigma, \delta, (1-\theta)\widetilde{\mathbf{P}}, \boldsymbol{\chi}, \mathscr{C}\big), \tag{13}$$

where $\widetilde{\mathbf{P}}$ is the the morph matrix of the underlying Markov chain.

As $\theta \to 0^+$, the resulting non-terminating plant model is denoted as

$$G(0) = (Q, \Sigma, \delta, \widetilde{\mathbf{P}}, \boldsymbol{\chi}, \mathscr{C}). \tag{14}$$

Definition 11: Given $\theta \in (0, 1)$, a terminating plant $G(\theta) = (Q, \Sigma, \delta, (1-\theta)\widetilde{\mathbf{P}}, \boldsymbol{\chi}, \mathscr{C})$ is defined to be the θ-neighbour of the non-terminating plant $G(0) = (Q, \Sigma, \ \delta, \widetilde{\mathbf{P}}, \boldsymbol{\chi}, \mathscr{C})$.

For a given non-terminating plant $G(0)$ and a fixed $\theta_0 \in (0, 1)$, there is exactly one θ_0-neighbour $G(\theta_0)$.

Notation 2: Let $\theta \in (0, 1)$ be the unform probability of termination for a terminating plant $G(\theta) = (Q, \Sigma, \delta, (1-\theta)\widetilde{\mathbf{P}}, \boldsymbol{\chi}, \mathscr{C})$. Let $\mathbf{P}$ be the state transition probability matrix of the underlying Markov chain, which is generated from δ and $\widetilde{\Pi}$ (see equation (2)). Then, the (renormalized) language measure vector (see Definition 8) is obtained as

$$\boldsymbol{\nu}(\theta) = \theta\Big[\mathbb{I} - (1-\theta)\mathbf{P}\Big]^{-1}\boldsymbol{\chi} \tag{15}$$

where $(1-\theta)\mathbf{P}$ is the sub-stochastic transition probability matrix for the terminating plant. Similarly, for a non-terminating plant $G(0) = (Q, \Sigma, \delta, \widetilde{\mathbf{P}}, \boldsymbol{\chi}, \mathscr{C})$ having the stochastic transition probability matrix $\mathbf{P}$, the (renormalized) measure vector (Chattopadhyay and Ray 2006a) is denoted as

$$\boldsymbol{\nu}(0) = \lim_{\theta \to 0^+} \boldsymbol{\nu}(\theta) = \lim_{\theta \to 0^+} \theta\Big[\mathbb{I} - (1-\theta)\mathbf{P}\Big]^{-1}\boldsymbol{\chi} \tag{16}$$

In the sequel, renormalized measure $\boldsymbol{\nu}$ in equations (10) and (11) is referred to as measure for brevity.

3.2 *Construction of an optimal supervisor*

A supervisor disables a subset of the set $\mathscr{C}$ of controllable transitions and hence there is a bijection between the set of all possible supervision policies and the power set $2^{\mathscr{C}}$. That is, there exists $2^{|\mathscr{C}|}$ possible supervisors and each supervisor is uniquely identifiable with a subset of $\mathscr{C}$ and the language measure ν allows a quantitative comparison of different supervision policies.

Definition 12: For an unsupervised (non-terminating) plant $G(0) = (Q, \Sigma, \delta, \widetilde{\mathbf{P}}, \boldsymbol{\chi}, \mathscr{C})$, let $G^\dagger$ and $G^\ddagger$ be the supervised plants with sets of disabled transitions, $\mathscr{D}^\dagger \subseteq \mathscr{C}$ and $\mathscr{D}^\ddagger \subseteq \mathscr{C}$, respectively, whose measures are $\boldsymbol{\nu}^\dagger$ and $\boldsymbol{\nu}^\ddagger$. Then, the supervisor that disables $\mathscr{D}^\dagger$ is defined to be superior to the supervisor that disables $\mathscr{D}^\ddagger$ if $\boldsymbol{\nu}^\dagger \geqq_{\mathbf{E}} \boldsymbol{\nu}^\ddagger$ and strictly superior if $\boldsymbol{\nu}^\dagger >_{\mathbf{E}} \boldsymbol{\nu}^\ddagger$.

Definition 13 (Optimal supervision problem)**:** Given a (non-terminating) plant $G(0) = (Q, \Sigma, \delta, \widetilde{\mathbf{P}}, \boldsymbol{\chi}, \mathscr{C})$, the problem is to compute a supervisor that disables a subset $\mathscr{D}^\star \subseteq \mathscr{C}$, such that

$$\boldsymbol{\nu}^\star \geqq_{\mathbf{E}} \boldsymbol{\nu}^\dagger \quad \forall \mathscr{D}^\dagger \subseteq \mathscr{C}$$

where $\boldsymbol{\nu}^\star$ and $\boldsymbol{\nu}^\dagger$ are the measure vectors of the supervised plants $G^\star$ and $G^\dagger$ under $\mathscr{D}^\star$ and $\mathscr{D}^\dagger$, respectively.

Remark 2: For a non-trivial plant $G(0) = (Q, \Sigma, \delta, \widetilde{\mathbf{P}}, \boldsymbol{\chi}, \mathscr{C})$ (i.e., $|Q| > 1$), there may exist two supervisors that are not comparable in the sense of Definition 12. For example, given a two-state unsupervised plant G, if $G^\dagger$ and $G^\ddagger$ are supervised plants under two different supervisors with disabled transition sets, $\mathscr{D}^\dagger$ and $\mathscr{D}^\ddagger$, respectively, then the following situation may occur for the indices $i \neq j$.

$$\left(\nu_i^\dagger > \nu_i^\ddagger\right) \bigwedge \left(\nu_j^\dagger < \nu_j^\ddagger\right),$$

where $\boldsymbol{\nu}_i^\dagger$ and $\boldsymbol{\nu}_i^\ddagger$ are the ith elements of the measure vectors for $G^\dagger$ and $G^\ddagger$, respectively. It is shown in the next section that, for a given plant, an optimal supervisor (in the sense of Definition 13) does exist for which the measure vector is elementwise greater than or equal to the measure vector of the plant under any other supervision policy.

Terminating plant models have sub-stochastic transition probability matrices (see Definition 7). By postulating the existence of unmodelled transitions, such plants can be transformed to non-terminating models as explained below. For uniform termination probability $\theta \in (0, 1)$, equations (8) and (11) suggest the possibility of computing optimal supervision policies for terminating plants based on the analysis of non-terminating plants.

4. Optimal control problem: solution

This section presents a solution to the optimal supervision problem by assuming a uniform non-zero probability of termination, θ, at each state. Then, it is shown that the solution for the corresponding non-terminating plant can be obtained from the control policy of the terminating plant and the bounds on the derivatives of the language measure (see Appendix A).

Let $\theta \in (0, 1)$ be the uniform termination probability of an unsupervised plant $G(\theta) = (Q, \Sigma, \delta, (1-\theta)\widetilde{\mathbf{P}}, \boldsymbol{\chi}, \mathscr{C})$. The resulting (substochastic) state transition cost matrix is $\boldsymbol{\Pi}(\theta) = (1-\theta)\mathbf{P}$. For such plants with uniform non-zero termination probability, the following lemma states existence of an augmented plant model.

Lemma 3: For the terminating plant $G(\theta) = (Q, \Sigma, \delta, (1-\theta)\widetilde{\mathbf{P}}, \boldsymbol{\chi}, \mathscr{C})$, let the corresponding augmented non-terminating plant be $G_{\text{aug}} = (Q_{\text{aug}}, \Sigma_{\text{aug}}, \delta_{\text{aug}}, \widetilde{\boldsymbol{\Pi}}_{\text{aug}}, \boldsymbol{\chi}_{\text{aug}}, \mathscr{C})$. Let $\boldsymbol{v}^\star(\theta)$ and $\boldsymbol{v}^\dagger(\theta)$ be the measures of the terminating plant with the respective sets of disabled transitions $\mathscr{D}^\star \subseteq \mathscr{C}$ and $\mathscr{D}^\dagger \subseteq \mathscr{C}$. Then,

$$\exists \mathscr{D}^\star \subseteq \mathscr{C} \quad \text{s.t. } \boldsymbol{v}^\star(\theta) \geqq_{\mathbf{E}} \boldsymbol{v}^\dagger(\theta) \quad \forall \mathscr{D}^\dagger \subseteq \mathscr{C} \quad \forall \theta \in (0,1) \tag{17}$$

which implies that an optimal supervisor for G_{aug} exists (in the sense of Definition 13) which disables $\mathscr{D}^\star \subseteq \mathscr{C}$.

Proof: The first n elements of the measure vectors of the augmented plant and the unaugmented plant are identically equal as seen in equation (11). Then, the proof follows from Definition 12. □

The remainder of this section derives an algorithm for a supervision policy that elementwise maximizes the measure of the terminating plant $G(\theta)$. Lemma 3 guarantees that the optimal policy is based on a non-terminating plant.

Proposition 2 (Monotonicity)**:** Let $\boldsymbol{\Pi}(\theta)$ and $\boldsymbol{v}(\theta)$ be the state transition cost matrix and the measure vector of an unsupervised plant $G(\theta) = (Q, \Sigma, \delta, (1-\theta)\widetilde{\mathbf{P}}, \boldsymbol{\chi}, \mathscr{C})$, respectively. Let a supervisor be constructed to reconfigure the plant by disabling a set of controllable transitions $\mathscr{D}^\dagger \subseteq \mathscr{C}$ such that $\boldsymbol{\Pi}$ is modified to $\boldsymbol{\Pi}^\dagger$ by following Algorithm 1. Then, denoting the measure vector of the supervised plant by $\boldsymbol{v}^\dagger$, it follows that $\boldsymbol{v}^\dagger \geqq_{\mathbf{E}} \boldsymbol{v}$; and equality holds if and only if $\boldsymbol{\Pi}^\dagger = \boldsymbol{\Pi}$.

Algorithm 1: Monotonic Increase of Measure $\boldsymbol{v}$

begin
  for $i = 1$ ***to*** n **do**
    for $j = 1$ ***to*** n **do**

$$\left.\begin{aligned} \Pi^\dagger_{ij} &= \Pi_{ij} + \beta_{ij} \\ \Pi^\dagger_{ii} &= \Pi_{ii} - \beta_{ij} \end{aligned}\right\} \text{ if } \boldsymbol{v}_j > \boldsymbol{v}_i \text{ with } \beta_{ij} > 0$$

$$\left.\begin{aligned} \Pi^\dagger_{ij} &= \Pi_{ij} \\ \Pi^\dagger_{ii} &= \Pi_{ii} \end{aligned}\right\} \text{ if } \boldsymbol{v}_j = \boldsymbol{v}_i$$

$$\left.\begin{aligned} \Pi^\dagger_{ij} &= \Pi_{ij} - \beta_{ij} \\ \Pi^\dagger_{ii} &= \Pi_{ii} + \beta_{ij} \end{aligned}\right\} \text{ if } \boldsymbol{v}_j < \boldsymbol{v}_i \text{ with } \beta_{ij} > 0$$

    endfor
  endfor
end

Proof: It follows from equation (15) in Notation 2 that

$$\begin{aligned} \boldsymbol{v}^\dagger - \boldsymbol{v} &= \theta\left[\mathbb{I} - \boldsymbol{\Pi}^\dagger\right]^{-1} - \theta[\mathbb{I} - \boldsymbol{\Pi}]^{-1}\boldsymbol{\chi} \\ &= \theta\left[\mathbb{I} - \boldsymbol{\Pi}^\dagger\right]^{-1}\left([\mathbb{I} - \boldsymbol{\Pi}] - [\mathbb{I} - \boldsymbol{\Pi}^\dagger]\right)[\mathbb{I} - \boldsymbol{\Pi}]^{-}1\boldsymbol{\chi} \\ &= \theta\left[\mathbb{I} - \boldsymbol{\Pi}^\dagger\right]^{-1}\left(\boldsymbol{\Pi}^\dagger - \boldsymbol{\Pi}\right)\boldsymbol{v}. \end{aligned}$$

Defining the matrix $\boldsymbol{\Delta} \triangleq \boldsymbol{\Pi}^\dagger - \boldsymbol{\Pi}$, and the ith row of $\boldsymbol{\Delta}$ as Δ_i, it follows that

$$\Delta_i{}^T \boldsymbol{v} = \sum_j \Delta_{ij} v_j = \sum_j \theta\beta_{ij}\Gamma_{ij} \tag{18}$$

where

$$\Gamma_{ij} = \begin{cases} (v_i - v_j) & \text{if } v_i > v_j \\ 0 & \text{if } v_i = v_j \Longrightarrow \Gamma_{ij} \geqq 0 \ \forall i, j. \\ (v_j - v_i) & \text{if } v_i < v_j \end{cases}$$

Since $\sum_{i=1}^n \boldsymbol{\Pi}_{ij} = \sum_{i=1}^n \boldsymbol{\Pi}^\dagger_{ij}; \ \forall j, k$, it follows from non-negativity of $\boldsymbol{\Pi}$, that $[I - \boldsymbol{\Pi}^\dagger]^{-1} >_{\mathbf{E}} \mathbf{0}$. Since $\beta_i \ \geq 0 \ \forall i$, it follows that $\Delta_i{}^T \boldsymbol{v} \ \geq \ 0 \ \forall i \ \Rightarrow \boldsymbol{v}^\dagger \geqq_{\mathbf{E}} \boldsymbol{v}$. For $v_j \neq 0$ and Δ as defined above, $\Delta_i^T \boldsymbol{v}^k = 0$ if and only if $\Delta = 0$. Then, $\boldsymbol{\Pi}^\dagger = \boldsymbol{\Pi}$ and $\boldsymbol{v}^\dagger = \boldsymbol{v}$. □

Corollary 1: Under an identical situation to that assumed in the statement of Proposition 2, let the plant be reconfigured as given in Algorithm 2. Then, denoting the measure vector of the modified plant by $\boldsymbol{v}^\dagger$, it follows that $\boldsymbol{v}^\dagger \leqq_{\mathbf{E}} \boldsymbol{v}$; and equality holds if and only if $\boldsymbol{\Pi}^\dagger = \boldsymbol{\Pi}$.

Algorithm 2: Monotonic Decrease of Measure $\boldsymbol{v}$

begin
  for $i = 1$ ***to*** n **do**
    for $j = 1$ ***to*** n **do**

$$\left.\begin{aligned} \Pi^\dagger_{ij} &= \Pi_{ij} + \beta_{ij} \\ \Pi^\dagger_{ii} &= \Pi_{ii} - \beta_{ij} \end{aligned}\right\} \text{ if } \boldsymbol{v}_j < \boldsymbol{v}_i \text{ with } \beta_{ij} > 0$$

$$\left.\begin{aligned} \Pi^\dagger_{ij} &= \Pi_{ij} \\ \Pi^\dagger_{ii} &= \Pi_{ii} \end{aligned}\right\} \text{ if } \boldsymbol{v}_j = \boldsymbol{v}_i$$

$$\left.\begin{aligned} \Pi^\dagger_{ij} &= \Pi_{ij} - \beta_{ij} \\ \Pi^\dagger_{ii} &= \Pi_{ii} + \beta_{ij} \end{aligned}\right\} \text{ if } \boldsymbol{v}_j > \boldsymbol{v}_i \text{ with } \beta_{ij} > 0$$

    endfor
  endfor
end

Proof: The proof is similar to that of Proposition 26. □

Proposition 2 facilitates formulation of the algorithm for computing an optimal supervisor for plants with

uniform non-zero probability of termination at each state. Let the kth iteration of the algorithm compute a set $\mathscr{D}^{[k]} \subseteq \mathscr{C}$ of controllable transitions to be disabled in the sense of the control philosophy in Definition 9. The language measure vector computed in the kth iteration of the algorithm is denoted by $\boldsymbol{\nu}^{[k]}$. The algorithm terminates at the $(k+1)$th iteration if $\mathscr{D}^{[k]} = \mathscr{D}^{[k+1]}$, which is the optimal set of disabled transitions computed by the algorithm and is denoted by $\mathscr{D}^{\star}$. The algorithm is started with the unsupervised plant (i.e., with all controllable transitions enabled) and hence $\mathscr{D}^{[0]} = \emptyset$. A formal description is given in Algorithm 3.

Algorithm 3: Optimal Supervisor Algorithm

```
input  : Π(θ) = (1 − θ)P, χ, 𝒞
output: Optimal set of disabled transitions 𝒟*
begin
    Set 𝒟^[0] = ∅ ;                  /* Initial disabling set */
    Set k = 1 ;
    Set Terminate = false ;
    while (Terminate == false) do
        Compute ν^[k] ;
        for j = 1 to n do
            for k = 1 to n do
                Disable all controllable transitions q_k -σ-> q_j
                such that ν_j < ν_k ;
                Enable all controllable transitions q_k -σ-> q_j
                such that ν_j ≧ ν_k ;
            endfor
        endfor
        Compute 𝒟^[k] ;                  /* k^th disabling set */
        if 𝒟^[k] == 𝒟^[k−1] then
            Terminate = true ;
        else
            k = k + 1 ;
        endif
    endw
    𝒟* = 𝒟^[k] ;                     /* Optimal disabling set */
end
```

Proposition 3: Let $\boldsymbol{\nu}^{[k]}$ be the language measure vector computed in the kth iteration of Algorithm 3. The measure vectors computed by the algorithm form an elementwise non-decreasing sequence, i.e., $\boldsymbol{\nu}^{[k+1]} \geqq_{\mathbf{E}} \boldsymbol{\nu}^{[k]}\ \forall k$.

Proof: Let the state transition probability matrix in the kth iteration of Algorithm 3 be denoted by $\boldsymbol{\Pi}^{[k]}$. Then, the matrix $\boldsymbol{\Pi}^{[k+1]}$ is generated from $\boldsymbol{\Pi}^{[k]}$ by following the procedure as described in Proposition 2. Hence, $\boldsymbol{\nu}^{[k+1]} \geqq_{\mathbf{E}} \boldsymbol{\nu}^{[k]}$. □

Proposition 4 (effectiveness)**:** Algorithm 3 is an effective procedure (Hopcroft *et al.* 2001), i.e., it is guaranteed to terminate.

Proof: Let $G(\theta) = (Q, \Sigma, \delta, (1-\theta)\widetilde{\mathbf{P}}, \boldsymbol{\chi}, \mathscr{C})$ be the unsupervised plant and let $Card(\mathscr{C}) = \ell \in \mathbb{N}$. Denoting the set of all permutations of the vector $[1\ 2\ \cdots\ \ell]^T$ by $\mathscr{P}(\ell)$, a function $\xi: 2^{\mathscr{C}} \longrightarrow \mathscr{P}(\ell)$ is defined as

1. $\forall \mathscr{D}^{\dagger} \subseteq \mathscr{C}, \left(\nu^{\dagger}_{i_1} > \nu^{\dagger}_{i_2} > \cdots > \nu^{\dagger}_{i_n}\right)$
   $\Longrightarrow \left(\xi(\mathscr{D}^{\dagger}) = [i_1\ i_2 \cdots i_n]^T\right)$
2. $\left(\nu^{\dagger}_{i_s} = \nu^{\dagger}_{i_t}\right) \wedge (i_s > i_t)$
   $\Longrightarrow \left(\text{if } \xi(\mathscr{D}^{\dagger})_s = i_s \text{ and } \xi(\mathscr{D}^{\dagger})_t = i_t \text{ then } s > t\right)$.

Let $\boldsymbol{\nu}^{[k]}$ be the measure vector computed in the kth iteration of Algorithm 3. Then, $\boldsymbol{\nu}^{[k]} = \boldsymbol{\nu}^{[k+1]}$ implies that Algorithm 3 terminates in $k+1$ iterations according to its stopping rule.

Next let $\mathscr{D}^{[k_1]}$ and $\mathscr{D}^{[k_2]}$ be the disabling sets in iterations k_1 and k_2, respectively. If $\xi(\mathscr{D}^{[k_1]}) = \xi(\mathscr{D}^{[k_2]})$, then $\boldsymbol{\nu}^{[k_1+1]} = \boldsymbol{\nu}^{[k_2+1]}$. Since $\xi(\mathscr{D}^{[k_1]}) = \xi(\mathscr{D}^{[k_2]})$, it follows from the definition of ξ that if $\nu^{[k_1]}_i > \nu^{[k_1]}_j$, then $\nu^{[k_2]}_i \geqq \nu^{[k_2]}_j$. If $\nu^{[k_2]}_i > \nu^{[k_2]}_j$ then controllable transitions $q_i \to_{\sigma} q_j$ are disabled in both iterations k_1+1 and k_2+1. If $\nu^{[k_1]}_i = \nu^{[k_1]}_j$, then disabling or enabling controllable transitions $q_i \to_{\sigma} q_j$ does not affect the measure vector. Hence, it follows that $\boldsymbol{\nu}^{[k_1+1]}$ and $\boldsymbol{\nu}^{[k_2+1]}$ can be obtained by disabling the same set of controllable transitions, thus implying $\boldsymbol{\nu}^{[k_1+1]} = \boldsymbol{\nu}^{[k_2+1]}$. Since the measure vectors can repeat only at the final iteration, Algorithm 3 is guaranteed to terminate within $Card(\mathscr{P}(\ell)) = \ell!$ iterations. Therefore, effectiveness of Algorithm 3 is established. □

Next it is established that Algorithm 3 is correct in the sense that an optimal supervision policy is generated.

Proposition 5 (Optimality)**:** For a terminating plant $G(\theta) = (Q, \Sigma, \delta, (1-\theta)\widetilde{\mathbf{P}}, \boldsymbol{\chi}, \mathscr{C})$, the supervision policy computed by Algorithm 3 is optimal in the sense of Definition 13.

Proof: Let $G(\theta)$ have the state transition cost matrix $\boldsymbol{\Pi}$, measure $\boldsymbol{\nu}^{[0]}$, no disabled events, i.e., $\mathscr{D}^0 = \emptyset$. Let $G(\theta)$ be configured as the supervised plant $G^{\star}(\theta)$ by application of Algorithm 3 when it stops.

Let $G^{\dagger}$ be another configured plant distinct from $G^{\star}$. Let $\mathscr{D}^{\star} \subseteq \mathscr{C}$ and $\mathscr{D}^{\dagger} \subseteq \mathscr{C}$ be the respective sets of disabled transitions and $\nu^{\star}$ and $\nu^{\dagger}$ be the respective measures for $G^{\star}$ and $G^{\dagger}$; and $\mathscr{D}^{\star} \neq \mathscr{D}^{\dagger}$.

Let the following set differences be denoted as: $\triangle\mathscr{D} \triangleq \mathscr{D}^{\star} \setminus \mathscr{D}^{\dagger}$ and $\nabla\mathscr{D} \triangleq \mathscr{D}^{\dagger} \setminus \mathscr{D}^{\star}$. An application of Algorithm 3 yields

- $\forall i, j\ \nu^{\star}_i > \nu^{\star}_j \Longrightarrow$ all controllable transitions $q_i \underset{\sigma}{\to} q_j$ are disabled.
- $\forall i, j\ \nu^{\star}_i \leqq \nu^{\star}_j \Longrightarrow$ all controllable transitions $q_i \underset{\sigma}{\to} q_j$ are enabled.

To change the plant configuration from $G^\star$ to $G^\dagger$, all transitions in $\triangle\mathscr{D}$ are enabled and all transitions in $\nabla\mathscr{D}$ are disabled. Since any such change requires us to either disable a transition $q_i \rightarrow_\sigma q_j$ where $\nu_i^\star \le \nu_j^\star$ or enable a disabled transition $q_i \rightarrow_\sigma q_j$ where $\nu_i^\star > \nu_j^\star$, it follows from Corollary 1 that $\boldsymbol{\nu}^\dagger \leqq_{\mathbf{E}} \boldsymbol{\nu}^\star$.

Since $G^\dagger$ is an arbitrary configuration distinct from $G^\star$, it follows that $G^\star$ is an optimal supervision policy in the sense of Definition 13. □

In the reported work on discrete event control of non-probabilistic regular languages (e.g., (Ramadge and Wonham 1987)), the emphasis is on computing the maximally permissive supervisor in the sense that the supervised plant language is the supremal controllable sub-language of the specification. A similar approach is taken for probabilistic regular languages (Garg 1992a, b). In contrast, the measure-theoretic concept in this paper computes a policy that maximizes the elements of the language measure vector elementwise to find a supervisor with maximal performance. Proposition 5 shows that there exists at least one optimal supervisor. Now it is shown that the optimal supervisor computed by Algorithm 3 is unique in the sense of being maximally permissive among all policies that guarantee optimal performance of the supervised plant.

Proposition 6 (uniqueness)**:** Given an unsupervised plant $G(\theta)$, the optimal supervisor $G^\star(\theta)$, computed by Algorithm 3, is unique in the sense that it is maximally permissive among all possible supervision policies with optimal performance. That is, if $\mathscr{D}^\star$ and $\mathscr{D}^\dagger$ are the disabled transition sets, and $\boldsymbol{\nu}^\star$ and $\boldsymbol{\nu}^\dagger$ are the language measure vectors for $G^\star$ and an arbitrarily supervised plant $G^\dagger$, respectively, then

$$\boldsymbol{\nu}^\star \equiv_{\mathbf{E}} \boldsymbol{\nu}^\dagger \Longrightarrow \mathscr{D}^\star \subset \mathscr{D}^\dagger \subseteq \mathscr{C}. \tag{19}$$

Proof: If $G^\star$ and $G^\dagger$ are distinct, then $\mathscr{D}^\# \neq \mathscr{D}^\star$. Given $\boldsymbol{\nu}^\star \equiv_{\mathbf{E}} \boldsymbol{\nu}^\#$, let $G^\star$ be reconfigured to $G^\dagger$ by disabling and/or re-enabling appropriate controllable transitions. It follows from equation (18) that

$$\begin{aligned} 0 = \boldsymbol{\nu}^\dagger - \boldsymbol{\nu}^\star &= \left[\mathbb{I} - \boldsymbol{\Pi}^\dagger\right]^{-1}\left(\boldsymbol{\Pi}^\dagger - \boldsymbol{\Pi}^\star\right)\boldsymbol{\nu}^\star \\ &\Rightarrow \left(\boldsymbol{\Pi}^\dagger - \boldsymbol{\Pi}^\star\right)\boldsymbol{\nu}^\star = 0. \end{aligned} \tag{20}$$

The ith element of $\left(\boldsymbol{\Pi}^\dagger - \boldsymbol{\Pi}^\star\right)\boldsymbol{\nu}^\star$ is expressed as the finite sum of real numbers

$$0 = \left(\left(\boldsymbol{\Pi}^\dagger - \boldsymbol{\Pi}^\star\right)\boldsymbol{\nu}^\star\right)_i = \sum_{r=1}^{\tau} T_r^i, \tag{21}$$

where $0 \le \tau \le 2^{Card(\mathscr{C})}$ and each T_r^i is of the form:

$$T_r^i = \begin{cases} \alpha_r^i(\nu_i^\star - \nu_j^\star) > 0, & \text{if } T_r^i \text{ arises due to disabling} \\ & \quad q_i \underset{\sigma}{\rightarrow} q_j \text{ for some } q_j \in Q \\ \alpha_r^i(\nu_j^\star - \nu_i^\star) \ge 0, & \text{if } T_r^i \text{ arises due to enabling} \\ & \quad q_i \underset{\sigma}{\rightarrow} q_j \text{ for some } q_j \in Q \end{cases} \tag{22}$$

because each α_r^i represents event occurrence probabilities and hence are positive, and the the logic of disabling and re-enabling follows Algorithm 3. Therefore, it follows from equation (22) that $T_r^i = 0\ \forall r \in \{1, \ldots, \tau\}$.

Hence, it is necessary to re-enable controllable transitions $q_i \rightarrow_\sigma q_j$ and disable the self loop at q_i such that $\nu_i^\dagger = \nu_j^\dagger$ for reconfiguration from $G_\theta^\dagger$ to $G_\theta^\star$. Note that all such transitions are guaranteed to be enabled in $G_\theta^\star$ (see line 10 in Algorithm 3). Therefore, given $\boldsymbol{\nu}^\star \equiv_{\mathbf{E}} \boldsymbol{\nu}^\dagger$, it follows that $\mathscr{D}^\star \subseteq \mathscr{D}^\dagger$. That is, $G^\star(\theta)$ is unique for all $\theta \in (0,1)$ in the sense that the configured plant is maximally permissive among all other configurations that yield the same optimal measure $\boldsymbol{\nu}^\star(\theta)$. □

4.1 *Optimal control of non-terminating plants*

This section presents the optimal supervision problem for non-terminating plants (i.e., with termination probability $\theta = 0$ at each state) having the structure $G(0) = (Q, \Sigma, \delta, \widetilde{\mathbf{P}}, \boldsymbol{\chi}, \mathscr{C})$ and the corresponding stochastic transition probability matrix is $\mathbf{P}$. The rationale for working on a terminating plant, instead of the non-terminating plant is explained below.

By maximizing the measure $\boldsymbol{\nu}(\theta)$ for a given $\theta \in (0,1)$, an optimal control law can be derived based on the state transition cost matrix $\boldsymbol{\Pi}(\theta) = (1-\theta)\mathbf{P}$ of the supervised plant language and the originally assigned $\boldsymbol{\chi}$-vector. Such an optimal control law is sought to be θ-independent in the sense of having the same disabling set $\mathscr{D} \subseteq \mathscr{C}$ for a given range of θ, where θ might be restricted to be not too far away from 0^+. On the other hand, from the perspective of numerical stability and accuracy in computation of $\boldsymbol{\nu}(\theta)$ (see Definition 8), it is desirable to have a relatively large positive value of θ. The results derived in this section serve toward establishing upper bounds on θ for which the optimal control law should be θ-independent and the associated computation is numerically stable. The main objective is summarized below.

A uniform non-zero probability of termination $\theta_\star \in (0,1)$ is to be computed such that the terminating plant $G(\theta_\star)$ and the

non-terminating plant $G(0)$ shall have the same the disabling set $\mathscr{D} \subseteq \mathscr{C}$. However, in general, their measures could be different, i.e., $\boldsymbol{\nu}(\theta_\star) \neq_{\mathbf{E}} \boldsymbol{\nu}(0)$.

Proposition 7: Let $(1-\theta)\mathbf{P}$ and $\boldsymbol{\nu}(\theta)$ be the state transition cost matrix and the measure of the plant $G(\theta) = (Q, \Sigma, \delta, (1-\theta)\widetilde{\mathbf{P}}, \boldsymbol{\chi}, \mathscr{C})$. Then, for all $q_i, q_j \in Q$, there exists $\theta_{ij}^\star \in (0, 1]$ such that $\forall \theta \in (0, \theta_{ij}^\star)$, the sign of $\big(\nu_i(\theta) - \nu_j(\theta)\big)$ is fixed (i.e., positive, negative or zero); and $\theta_{ij}^\star$ can be computed as an explicit function of the stochastic matrix $\mathbf{P}$ and state characteristic vector $\boldsymbol{\chi}$.

Proof: Let $\gamma_{ij}(\theta) \triangleq \nu_i(\theta) - \nu_j(\theta)\ \forall \theta \in (0, 1)$, which is a smooth function of θ, and $\gamma_{ij}(0) = \lim_{\theta \to 0^+} \gamma_{ij}(\theta)$. The proof is based on the following two cases.

Case 1: No sign change of $\gamma_{ij}(\theta)$ in $(0, 1) \Rightarrow \theta_{ij}^\star = 1$. This includes: $\gamma_{ij}(0) = 0$ and $(d^k \gamma_{ij}(\theta)/d\theta^k)|_{\theta=0} = 0$ for all $k \geq 0$ because $\gamma_{ij}(\theta) = 0\ \forall \theta \in (0, 1)$ by Proposition A.3.

Case 2: $\gamma_{ij}(\theta)$ changes sign in $(0, 1)$; and $(\partial^r \gamma_{ij}(\theta)/\partial\theta^r)|_{\theta=0} = \lambda_{ij} \neq 0$ for some integer $r \geq 0$.

If $r = 0$, there exists $\tau_1 \in (0, 1)$ such that $\gamma_{ij}(\tau_1) = 0$ for the first time. If $r > 0$, it is possible that $\gamma_{ij}(0) = 0$. Then, as θ is increased from zero, $\gamma_{ij}(\theta)$ becomes non-zero and there exists $\tau_1 \in (0, 1)$ such that $\gamma_{ij}(\tau_1) = 0$ again. Smoothness of $\gamma_{ij}(\theta)$ necessitates that $(\partial^r \gamma_{ij}(\theta)/\partial\theta^r)|_{\theta=\theta_{ij}^\star} = 0$ for some $\theta_{ij}^\star \in (0, \tau_1)$. Then, it follows from the Mean value Theorem that there exists $\tau_2 \in (0, \theta_{ij}^\star)$ such that

$$\frac{\partial^{r+1}\gamma_{ij}(\theta)}{\partial\theta^{r+1}}\Big|_{\theta=\tau_2} = \frac{\lambda_{ij}}{\theta_{ij}^\star}$$

for the given $r \geq 0$, Proposition A.2, triangular inequality, and the relation $\gamma_{ij}(\theta) = \nu_i(\theta) - \nu_j(\theta)$ yield

$$\theta_{ij}^\star = \frac{\Big|(\partial^r \nu_i(\theta)/\partial\theta^r)\big|_{\theta=0} - (\partial^r \nu_j(\theta)/\partial\theta^{r+1})\big|_{\theta=0}\Big|}{(r+1)!\ 2^{r+3}\left(inf_{\alpha\neq 0}\left\|\left[\mathbb{I} - P + \alpha\mathscr{P}\right]^{-1}\right\|_\infty\right)^{r+1}}$$

$$= \begin{cases} \dfrac{|\{[\mathbb{I} - \mathbf{P} + \mathscr{P}]^{-1}\chi - \mathscr{P}\chi\}_i - \{[\mathbb{I} - \mathbf{P} + \mathscr{P}]^{-1}\chi - \mathscr{P}\chi\}_j|}{8 \times \left(inf_{\alpha\neq 0}\|[\mathbb{I} - \mathbf{P} + \alpha\mathscr{P}]^{-1}\|_\infty\right)}, & \text{if } r = 0 \\[2ex] \dfrac{\left|\begin{matrix} \left\{[\mathbb{I} - \mathbf{P} + \mathscr{P}]^{-1}\left[\mathbb{I} - [\mathbb{I} - \mathbf{P} + \mathscr{P}]^{-1}\right]^r \chi\right\}_i \\ -\left\{[\mathbb{I} - \mathbf{P} + \mathscr{P}]^{-1}\left[\mathbb{I} - [\mathbb{I} - \mathbf{P} + \mathscr{P}]^{-1}\right]^r \chi\right\}_j \end{matrix}\right|}{2^{r-3}\left(\inf_{\alpha\neq 0}\left\|\left[\mathbb{I} - P + \alpha\mathscr{P}\right]^{-1}\right\|_\infty\right)^{r+1}}, & \text{if } r > 0. \end{cases} \tag{23}$$

$\square$

Remark 3: For a non-terminating plant $G(0) = (Q, \Sigma, \delta, \widetilde{\mathbf{P}}, \boldsymbol{\chi})$, let $\theta_\star = \min_{i,j} \theta_{ij}^\star$. Then, the plant configuration obtained by applying a single iteration of Algorithm 3 to the θ-parameterized plant $G(\theta) = (Q, \Sigma, \delta, (1-\theta)\widetilde{\mathbf{P}}, \boldsymbol{\chi}, \mathscr{C})$ is identical for all $\theta \leq \theta_\star$.

The procedure of computing $\theta_\star$ is summarized as Algorithm 4.

Proposition 8: Complexity of computing a positive bound for $\theta_\star$ is $O(n^3)$ where n is the number of plant states.

Algorithm 4: Computation of the Bound $\theta_\star$

input : $\mathbf{P}$, χ
output: $\theta_\star$

begin
    Set $\theta_\star = 1$;
    Set $\theta_{curr} = 0$;
    Compute $\mathscr{P}$;   `/* See Algorithm 6 in Appendix A */`
    Compute $M_0 = \left[\mathbb{I} - \mathbf{P} + \mathscr{P}\right]^{-1}$;
    Compute $M_1 = \left[\mathbb{I} - \left[\mathbb{I} - \mathbf{P} + \mathscr{P}\right]^{-1}\right]$;
    Compute $M_2 = inf_{\alpha\neq 0}\left\|\left[\mathbb{I} - \mathbf{P} + \alpha\mathscr{P}\right]^{-1}\right\|_\infty$;
    for $j = 1$ ***to*** n **do**
        for $i = 1$ ***to*** n **do**
            if $(\mathscr{P}\chi)_i - (\mathscr{P}\chi)_j \neq 0$ **then**
                $\theta_{curr} = \frac{1}{8M_2}\left|(\mathscr{P}\chi)_i - (\mathscr{P}\chi)_j\right|$
            else
                for $r = 0$ ***to*** n **do**
                    if $(M_0\chi)_i \neq (M_0\chi)_j$ **then**
                        Break;
                    else
                        if $\left(M_0 M_1^r \chi\right)_i \neq \left(M_0 M_1^r \chi\right)_j$ **then**
                            Break;
                        endif
                    endif
                endfor
                if $r == 0$ **then**
                    $\theta_{curr} = \frac{|(M_0 - \mathscr{P})\chi_i - (M_0 - \mathscr{P})\chi_j|}{8M_2}$;
                else
                    if $r > 0$ ***AND*** $r \leq n$ **then**
                        $\theta_{curr} = \frac{|(M_0 M_1 \chi)_i - (M_0 M_1 \chi)_j|}{2^{r+3} M_2}$;
                    else
                        $\theta_{curr} = 1$;
                    endif
                endif
            endif
            $\theta_\star = \min(\theta_\star, \theta_{curr})$;
        endfor
    endfor
end

Proof: Referring to Algorithm 4, the part within the nested For loops (lines 10 to 32) is executed at most n^2 times and each iteration involves only single-iteration scalar operations. Thus the computational complexity of this part is of the order of $O(n^2)$. Lines 5 and 6 involve inversion of $n \times n$ dimensional non-singular matrices and hence the complexity of execution is of the order of $O(n^3)$. Proposition G (see Appendix) guarantees that the complexity of computing $\mathscr{P}$ is, in general, of the order of $O(n^3)$. Line 7, which computes $M_2 = \inf_{\alpha \neq 0} \|[\mathbb{I} - \mathbf{P} + \alpha_0 \mathscr{P}]^{-1}\|_\infty$, is a search problem. However, since M_2 appears only in the denominator of the expressions for θ_{curr}, it follows that, if for some $\alpha = \alpha_0 \neq 0$ and by using

$$M_2 = \left\| \left[\mathbb{I} - \mathbf{P} + \alpha_0 \mathscr{P}\right]^{-1} \right\|_\infty \tag{24}$$

it is possible to obtain a positive lower bound of $\theta_\star$ in Algorithm 4. Since the computation of $\|[\mathbb{I} - \mathbf{P} + \alpha_0 \mathscr{P}]^{-1}\|_\infty$ is of the order of $O(n^3)$ due to the matrix inversion, it is concluded that a positive lower bound of $\theta_\star$ can be computed with a complexity of $O(n^3)$. □

Remark 4: It is shown in (Chattopadhyay and Ray 2006a) that for any stochastic matrix $\mathbf{P}$

$$\begin{aligned}\left[\mathbb{I} - \mathbf{P} + \alpha\mathscr{P}\right]^{-1} &= \left[\mathbb{I} - \mathbf{P} + \mathscr{P}\right]^{-1} \\ &\quad + \left(\frac{1-\alpha}{\alpha}\right)\mathscr{P} \quad \forall \alpha \neq 0 \\ &\Longrightarrow \left(\left[\mathbb{I} - \mathbf{P} + \mathscr{P}\right]^{-1} - \mathscr{P}\right) + \left(\frac{1}{\alpha}\right)\mathscr{P}. \end{aligned} \tag{25}$$

Using $M_2 = \|[\mathbb{I} - \mathbf{P} + \mathscr{P}]^{-1}\|_\infty$ instead of $M_2 = \inf_{\alpha \neq 0} \|[\mathbb{I} - \mathbf{P} + \alpha\mathscr{P}]^{-1}\|_\infty$ (i.e., using $\alpha = 1$) in Algorithm 4 yields a value which satisfies the requirement stated in Remark 3 and therefore qualifies as $\theta_\star$. Thus, the major advantage of this approximation is having significantly smaller computational complexity because the search involved in computing the infimum is avoided at the cost of using a smaller value of $\theta_\star$, which may make subsequent computation of measure slightly more difficult due to possible ill-conditioning (see Definition 8).

On account of Proposition 7 and Remark 3, Algorithm 3 is modified to solve the optimal supervision problem for non-terminating plants and the modified version is formally presented in Algorithm 5.

Algorithm 5: Computation of Optimal Supervisor

input : $\mathbf{P}$, χ, $\mathscr{C}$
output: Optimal set of disabled transitions $\mathscr{D}^\star$
begin
  Set $\mathscr{D}^{[0]} = \emptyset$; /* Initial disabling set */
  Set $\widetilde{\Pi}^{[0]} = \widetilde{\Pi}$; /* Initial event prob. matrix */
  Set $\theta_\star^{[0]} = 0.99$;
  Set $k = 1$;
  Set `Terminate` = **false** ;
  while (`Terminate` == **false**) **do**
    Compute $\theta_\star^{[k]}$; /* Algorithm 4 */
    Set $\widetilde{\Pi}^{[k]} = \frac{1-\theta_\star^{[k]}}{1-\theta_\star^{[k-1]}} \widetilde{\Pi}^{[k-1]}$;
    Compute $\nu^{[k]}$;
    for $j = 1$ ***to*** n **do**
      for $i = 1$ ***to*** n **do**
        Disable all controllable transitions $q_i \xrightarrow{\sigma} q_j$ such that $\nu_j^{[k]} < \nu_i^{[k]}$;
        Enable all controllable transitions $q_i \xrightarrow{\sigma} q_j$ such that $\nu_j^{[k]} \geqq \nu_i^{[k]}$;
      endfor
    endfor
    Collect all disabled transitions in $\mathscr{D}^{[k]}$;
    if $\mathscr{D}^{[k]} == \mathscr{D}^{[k-1]}$ **then**
      `Terminate` = **true** ;
    else
      $k = k + 1$;
    endif
  endw
  $\mathscr{D}^\star = \mathscr{D}^{[k]}$; /* Optimal disabling set */
end

Proposition 9 (effectiveness)**:** Algorithm 5 is an effective procedure (Hopcroft *et al.* 2001), i.e., it is guaranteed to terminate.

Proof: Comparison of Algorithm 3 and Algorithm 5 reveals that while the former assumes a fixed probability of termination θ at each state, the latter modifies this parameter, denoted as $\theta_\star^{[k]}$, at each iteration k. Let $\theta_{\min} = \min\left(\theta_\star^{[1]}, \theta_\star^{[2]}\right)$ and let $\mathscr{D}^{[1]}(\theta_{\min})$ and $\mathscr{D}^{[2]}(\theta_{\min})$ be sets of disabled transition at the first and second iterations, respectively, for the terminating plant $G(\theta_{\min})$. Similarly, for the non-terminating plant $G(0)$, let $\mathscr{D}^{[1]}(0)$ and $\mathscr{D}^{[2]}(0)$ be the sets of disabled transitions at the first and second iterations, respectively. It follows from Remark 3 that $\mathscr{D}^{[1]}(0) = \mathscr{D}^{[1]}(\theta_{\min})$ and $\mathscr{D}^{[2]}(0) = \mathscr{D}^{[2]}(\theta_{\min})$.

Extending the above argument by induction based on k iterations of Algorithm 5 and denoting $\theta_{\min} = \min(\theta_\star^{[1]}, \ldots, \theta_\star^{[k]})$, an application of Algorithm 3 on a terminating plant $G(\theta_{\min})$ yields

$$\mathscr{D}^{[r]}(0) = \mathscr{D}^{[r]}(\theta_{\min}) \; \forall r \in \{1, \ldots, k\}.$$

Proposition 4 states that, for an arbitrary plant, Algorithm 3 is guaranteed to terminate within finitely many iterations. Hence, Algorithm 5 is an effective procedure. □

Next, it is shown that the plant configuration obtained by Algorithm 5 is optimal in the sense of Definition 13.

Proposition 10 (optimality)**:** For a non-terminating plant $G(0) = (Q, \Sigma, \delta, \widetilde{\mathbf{P}}, \chi, \mathscr{C})$), the supervision policy computed by Algorithm 5 is optimal in the sense of Definition 13.

Proof: Let the set of disabled transitions computed at the kth iteration Algorithm 5 be denoted by $\mathscr{D}_{\lim}^{[k]}$ and the termination probability be denoted by $\theta_\star^{[k]}$. Let the set of disabled transitions at the convergence of Algorithm 5 be $\mathscr{D}_{\lim}^{[m]}$. Let $\theta_{\min} = \min_{r\in\{1,\ldots,\ell\}} (\theta_\star^{[1]}, \ldots, \theta_\star^{[\ell]}) > 0$.

Let $G(\theta_{\min})$ be a terminating plant with $\boldsymbol{\Pi}(\theta_{\min}) = (1 - \theta_{\min})\mathbf{P}$. It follows from the proof of Proposition 9 that applications of Algorithm 3 to $G(\theta_{\min})$ and Algorithm 5 to $G(0)$ yield the same set $\mathscr{D}$ of disabled controllable events although the optimal measures, being θ-dependent would be different, i.e., $\boldsymbol{v}(\theta_{\min}) \neq_{\mathbf{E}} \boldsymbol{v}(0)$.

Proposition 5 implies that the optimal disabling set for a plant $G(\theta)$ generates the the same set of disabled controllable transitions for all $0 < \theta \leqq \theta_{\min}$. Because of continuity of $\boldsymbol{v}(\theta)$ with respect to θ, it is argued that $G^\star(0)$ is optimal in the sense of Definition 13, i.e., $\boldsymbol{v}^\star \geqq_{\mathbf{E}} \boldsymbol{v}^\dagger$, where $G^\dagger(0)$ is obtained by arbitrarily disabling controllable transitions in G. This completes the proof. □

Next it is shown that the supervision policy computed by Algorithm 5 is unique in the same sense as Proposition 6.

Proposition 11 (Uniqueness)**:** Let $G(0)$ be an unsupervised non-terminating plant and $G^\star(0)$ be the supervised plant configured by Algorithm 5. Then, $G^\star$ is unique in the sense that it is maximally permissive among supervised plants that yield optimal performance based on θ-neighbours $G(\theta)$ of $G(0)$ (see Definition 11) for all $\theta \in (0, \theta_\star)$, where $\theta_\star$ is computed by Algorithm 4. Equivalently, if $G^\dagger(\theta)$ is an arbitrarily supervised plant, then the following condition holds:

$$\left(\boldsymbol{v}^\star(\theta) \geqq_{\mathbf{E}} \boldsymbol{v}^\dagger(\theta)\right) \bigwedge \left(\left(\mathscr{D}^\star \subseteq \mathscr{D}^\dagger\right) \bigvee \left(\boldsymbol{v}^\star(\theta) \neq_{\mathbf{E}} \boldsymbol{v}^\dagger(\theta)\right)\right),$$

where $\boldsymbol{v}$ and $\mathscr{D}$ denote respective language measures and sets of disabled transitions.

Proof: It follows from Proposition 10 that $\boldsymbol{v}^\star(0) \geqq_{\mathbf{E}} \boldsymbol{v}^\dagger(0)$. It also follows from Proposition that $\boldsymbol{v}^\star(\theta) \geqq_{\mathbf{E}} \boldsymbol{v}^\dagger(\theta)$ for $\theta \in (0, \theta_\star)$. If $\boldsymbol{v}^\star(\theta) \equiv_{\mathbf{E}} \boldsymbol{v}^\dagger(\theta)$, then $G^\star(\theta)$ and $G^\dagger(\theta)$ are both optimal supervised configurations of the unsupervised terminating plant $G(\theta)$. It follows from Proposition 6 that $\mathscr{D}^\star \subseteq \mathscr{D}^\dagger$; otherwise $\boldsymbol{v}^\star(\theta) \neq_{\mathbf{E}} \boldsymbol{v}^\dagger(\theta)$. □

Proposition 12: Computational complexity of Algorithm 5 is of the same order as that of Algorithm 3.

Proof: Algorithm 5 computes $\theta_\star$ in each iteration and complexity of this computation is $O(n^3)$, where n is the number of states in the plant (see Proposition 8). Each iteration of both Algorithm 3 and Algorithm 5 involves computation of the measure vector $\boldsymbol{v}$, whose complexity is also $O(n^3)$ because of $n \times n$ matrix inversion. Hence, computational complexity of each iteration is $O(n^3)$ for both Algorithm 3 and Algorithm 5. Finally, the argument presented in Proposition 9 implies that the number of iterations in Algorithm 5 is of the same order as that in Algorithm 3. This completes the proof. □

4.2 *Testing of computational complexity*

Proposition 4 shows that Algorithm 3 is an effective procedure (Hopcroft *et al.* 2001), i.e., the solution is guaranteed to converge in a finite number of iterations. Extensive simulation suggests that the the maximum number of iterations for Algorithm 3 is actually of polynomial order in n, where n is the number of states in the unsupervised plant. The result is illustrated in figure 6, where the maximum number of required iterations $I_{\max}$ is plotted against number, n, of plant states. For each n, 10,000 simulation runs were conducted for synthesis of optimal plant configuration with randomly generated entries in the pair $\big((1 - \theta)\mathbf{P},\ \chi\big)$; and $I_{\max}$ was chosen to be the maximum number of iterations required by Algorithm 3 to converge; this is the most conservative case. The plot in figure 1 shows a distinct sub-linear variation. The following conjecture is made based on these observations.

Conjecture 1 (polynomial convergence)**:** *Given a terminating plant $G(\theta)$ with a uniform non-zero probability of termination θ at each of the n plant states,*

1. *Algorithm 3 converges in at most $O(n)$ iterations.*
2. *Computational complexity of Algorithm 3 is bounded by $O(n^4)$.*

Statement 2 in Conjecture 1 follows from Statement 1 and the following facts: Each iteration has complexity of $O(n^3)$ due to matrix inversion in the computation of

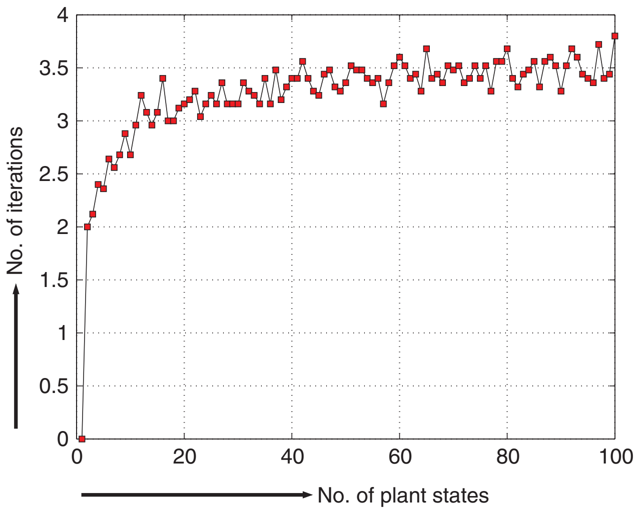


Figure 1. Number of iterations to converge in Algorithm 3.

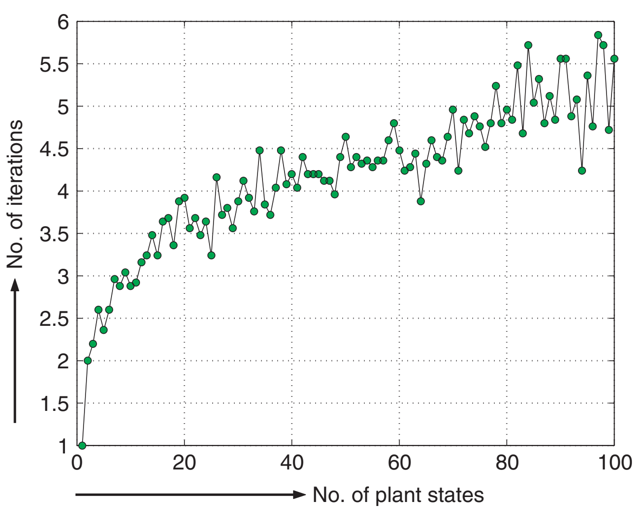


Figure 2. Number of iterations to converge in Algorithm 5.

the language measure vector, and matrix inversion has complexity of $O(n^3)$). Combination of Conjecture 1 and Proposition 12 implies that Algorithm 5 converges in $O(n)$ iterations and that complexity of the algorithm is $O(n^4)$. Similar to the procedure, described above for Algorithm 3, 10,000 random simulation runs for each n were conducted for testing Algorithm 5. Figure 2 shows the plot of average number of iterations required to converge at each value of n in contrast to figure 1, where the maximum number of iterations is potted. As expected, the plot of figure 2 is also sub-linear.

5. Optimal control of three processor message decoding

This section presents the design of a discrete-event (controllable) supervisor for a multiprocessor message

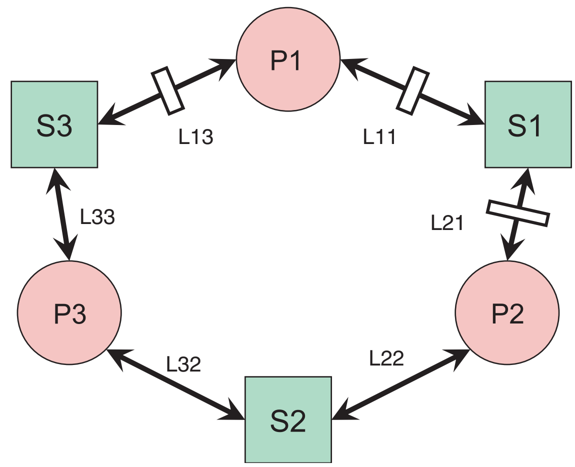


Figure 3. Arrangement of the processor links.

decoding system, described in an earlier publication (Ray *et al.* 2004). The optimal supervisory algorithm has been synthesized based on the algorithms presented in earlier sections.

Figure 3 depicts the arrangement of the message decoding system, where each of the three processors, p_1, p_2 and p_3, receives encoded messages that are to be decoded. The processor p_3 normally receives the most important messages, and p_1 receives the least important messages. There is a server between each pair of processors—s_1 between p_1 and p_2; s_2 between p_2 and p_3; and s_3 between p_3 and p_1. Each server is connected to each of its two adjacent processors by a link—the server s_j is connected to the adjacent processors p_i and p_k through the links L_{ij} and L_{kj}, respectively. Out of these six links, each of the three links, L_{11}, L_{12}, and L_{21}, is equipped with a switch to disable the respective connection whenever it is necessary; each of the remaining three links, L_{22}, L_{32}, and L_{33}, always remain connected. Each server s_i is equipped with a decoding key k_i that, at any given time, can only be accessed by only one of the two processors, adjacent to the server, through the link connecting the processor and the server. In order to decode the message, the processor holds the information on both keys of the servers next to it, one at a time. After decoding, the processor simultaneously releases both keys so that other processors may obtain access to them.

Figure 4 depicts the unsupervised plant model of the decoding system as a finite state automaton, where state 1 is the initial state. The event p_{ij} indicates that processor p_i has accessed the key k_j; and the event f_i indicates that the processor p_i has finished decoding and (simultaneously) released both keys in its possession upon completion of decoding. The events f_i are uncontrollable because, after the decoding is

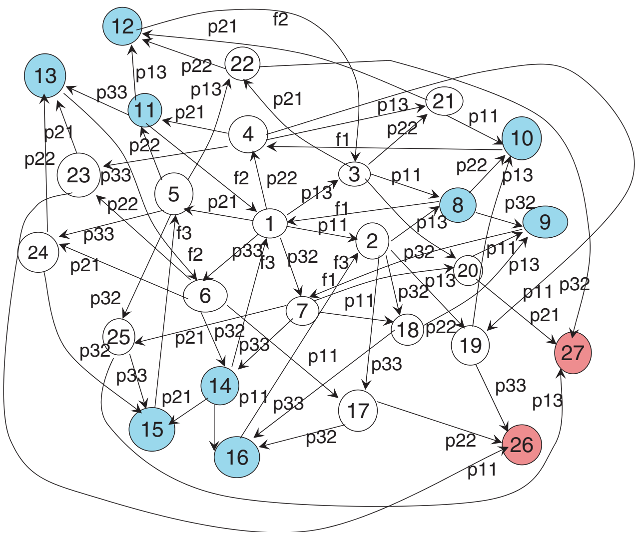


Figure 4. Finite state model of the message decoding system.

initiated, there is no control on when a processor finishes decoding.

Table 1 lists the event cost matrix $\widetilde{\mathbf{\Pi}}$. Two different control specifications are investigated. The first set of specifications, which emphasizes avoiding deadlock, is represented by the $\boldsymbol{\chi}$ vector in the first column of table 2. The second set of specifications, which focuses on increasing the throughput of processor 1, is represented by the $\boldsymbol{\chi}$ vector in the second column of table 2. The positive elements of the $\boldsymbol{\chi}$ vector are assigned to the states 8 to 16 that represent successful decoding of each processor. The $\boldsymbol{\chi}$ values of the deadlock states 26 and 27, where each processor holds exactly one key and hence no processor releases its key, are assigned negative values. The remaining states are non-marked and are assigned zero weights.

Algorithm 5 is applied to obtain the sequence of measure vectors for the two control specifications. The results of successive iterations, enumerating the renormalized measure vectors, are presented in Table 3 and 4 respectively. The last column in each table is the optimal renormalized measure vector. The optimization requires 7 iterations in Case 1 and 5 iterations in Case 2.

The optimal configurations for the plant obtained under Algorithm 5 are depicted in figures 5 and 6 respectively. For supervisor policy 1, the controlled plant is not trim and, for supervisor policy 2, there are disconnected states in the controlled model. This is interpreted as the supervisor successfully preventing the plant from visiting these states. The critical values for the termination probability $\theta_\star$ computed by the optimization algorithm for each control specification is shown in figure 7.

Table 1. Event occurrence probabilities for processor models.

$p11$	$p13$	$p21$	$p22$	$p32$	$p33$	$f1$	$f2$	$f3$
0.16	0.04	0.16	0.16	0.16	0.32	0.00	0.00	0.00
0.00	0.16	0.00	0.26	0.26	0.32	0.00	0.00	0.00
0.37	0.00	0.21	0.21	0.21	0.00	0.00	0.00	0.00
0.32	0.11	0.26	0.00	0.00	0.32	0.00	0.00	0.00
0.00	0.11	0.00	0.28	0.28	0.33	0.00	0.00	0.00
0.25	0.00	0.25	0.25	0.25	0.00	0.00	0.00	0.00
0.28	0.11	0.28	0.00	0.00	0.33	0.00	0.00	0.00
0.00	0.00	0.00	0.39	0.39	0.00	0.22	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
0.00	0.14	0.00	0.00	0.00	0.43	0.00	0.43	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
0.33	0.00	0.33	0.00	0.00	0.00	0.00	0.00	0.34
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00
0.00	0.00	0.00	0.50	0.50	0.00	0.00	0.00	0.00
0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.25	0.00	0.00	0.00	0.75	0.00	0.00	0.00
0.50	0.00	0.50	0.00	0.00	0.00	0.00	0.00	0.00
0.50	0.00	0.50	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.50	0.50	0.00	0.00	0.00	0.00
0.50	0.00	0.50	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.50	0.50	0.00	0.00	0.00	0.00
0.00	0.25	0.00	0.00	0.00	0.75	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.33	0.33	0.34
0.00	0.00	0.00	0.00	0.00	0.00	0.33	0.33	0.34

Table 2. Vectors for control specifications.

Case 1				Case 2	
0.000	0.010	0.000	0.000	1.000	0.000
0.000	0.020	0.000	0.000	0.020	0.000
0.000	0.020	0.000	0.000	0.020	0.000
0.000	0.020	0.000	0.000	0.020	0.000
0.000	0.040	0.000	0.000	0.040	0.000
0.000	0.040	0.000	0.000	0.040	0.000
0.000	0.040	0.000	0.000	0.040	0.000
0.010	0.000	−1.000	1.000	0.000	−0.200
0.010	0.000	−1.000	1.000	0.000	−0.200

Next the stable probability distributions of the plant states are compared for the following three cases:

- Open-loop or unsupervised plant
- Plant with the optimal supervision policy for specification 1
- Plant with the optimal supervision policy for specification 2

The distributions are obtained by considering the first row of the matrix $\mathscr{P}$, based on the measure ν_1

Table 3. Iteration vectors for multi-processor model: case 1.

Itr 1	Itr 2	Itr 3	Itr 4	Itr 5	Itr 6	Itr 7
−0.0616	0.0006	0.0110	0.0124	0.0124	0.0143	0.0143
−0.0616	0.0001	0.0063	0.0124	0.0124	0.0143	0.0143
−0.0616	0.0003	0.0055	0.0124	0.0124	0.0143	0.0143
−0.0616	0.0007	0.0110	0.0124	0.0124	0.0143	0.0143
−0.0616	0.0011	0.0110	0.0124	0.0124	0.0143	0.0143
−0.0616	0.0012	0.0110	0.0124	0.0124	0.0143	0.0143
−0.0616	0.0000	0.0093	0.0124	0.0124	0.0143	0.0143
−0.0616	0.0002	0.0110	0.0124	0.0124	0.0143	0.0143
−0.0616	0.0000	0.0093	0.0124	0.0124	0.0143	0.0143
−0.0616	0.0007	0.0110	0.0124	0.0124	0.0143	0.0143
−0.0616	0.0007	0.0110	0.0124	0.0124	0.0143	0.0143
−0.0616	0.0003	0.0055	0.0124	0.0124	0.0143	0.0143
−0.0616	0.0013	0.0110	0.0124	0.0124	0.0143	0.0143
−0.0616	0.0010	0.0110	0.0124	0.0124	0.0143	0.0143
−0.0616	0.0011	0.0110	0.0124	0.0124	0.0143	0.0143
−0.0616	0.0001	0.0063	0.0124	0.0124	0.0143	0.0143
−0.0616	0.0001	0.0000	0.0124	0.0124	0.0143	0.0143
−0.0616	0.0000	0.0093	0.0124	0.0124	0.0143	0.0143
−0.0616	0.0009	0.0110	0.0124	0.0124	0.0143	0.0143
−0.0616	0.0000	0.0000	0.0124	0.0124	0.0143	0.0143
−0.0616	0.0003	0.0110	0.0124	0.0124	0.0143	0.0143
−0.0616	0.0003	0.0000	0.0124	0.0124	0.0143	0.0143
−0.0616	0.0014	0.0110	0.0124	0.0124	0.0143	0.0143
−0.0616	0.0012	0.0110	0.0124	0.0124	0.0143	0.0143
−0.0616	0.0012	0.0110	0.0124	0.0124	0.0143	0.0143
−0.0616	0.0007	0.0110	0.0124	0.0124	0.0143	0.0143
−0.0616	0.0007	0.0110	0.0124	0.0124	0.0143	0.0143

Table 4. Iteration vectors for multi-processor model: case 2.

Itr 1	Itr 2	Itr 3	Itr 4	Itr 5
0.0598	0.2076	0.2879	0.3245	0.3245
0.0598	0.2074	0.2880	0.3245	0.3245
0.0598	0.2101	0.2879	0.3245	0.3245
0.0598	0.2167	0.2876	0.3232	0.3245
0.0598	0.2109	0.2878	0.3236	0.3245
0.0598	0.2084	0.2882	0.3245	0.3245
0.0598	0.2059	0.2878	0.3245	0.3245
0.0598	0.2090	0.2879	0.3245	0.3245
0.0598	0.2059	0.2878	0.3245	0.3245
0.0598	0.2175	0.2875	0.3230	0.3245
0.0598	0.2089	0.2879	0.3245	0.3245
0.0598	0.2105	0.2879	0.3245	0.3245
0.0598	0.2086	0.2882	0.3245	0.3245
0.0598	0.2078	0.2879	0.3245	0.3245
0.0598	0.2114	0.2878	0.3235	0.3245
0.0598	0.2076	0.2880	0.3245	0.3245
0.0598	0.2080	0.2880	0.3245	0.3245
0.0598	0.2059	0.2878	0.3245	0.3245
0.0598	0.2216	0.2872	0.3241	0.3245
0.0598	0.2059	0.2878	0.3245	0.3245
0.0598	0.2147	0.2879	0.3245	0.3245
0.0598	0.2116	0.2879	0.3245	0.3245
0.0598	0.2084	0.2879	0.3245	0.3245
0.0598	0.2105	0.2878	0.3232	0.3245
0.0598	0.2110	0.2878	0.3236	0.3245
0.0598	0.2077	0.2879	0.3245	0.3245
0.0598	0.2077	0.2879	0.3245	0.3245

corresponding to state 1 which is the initial state in both cases. If the stochastic matrix **P** is primitive (i.e., irreducible and acyclic), then all rows of $\mathscr{P}$ would be identical. However, primitiveness of **P** is not guaranteed even if the unsupervised plant model have this property because any subsequent event disabling may cause loss of reducibility or acyclic properties.

The results on evolution of the distribution are plotted in figure 8. While the unsupervised plant has a finite probability of reaching the deadlock states 26 and 27, the optimal supervisors in both cases successfully prevent occurrence of deadlock in the sense that the stable occupation probabilities for states 26 and 27 are zero for each supervisor. However, supervisor 2 increases the throughput of processor 1 as seen from the increased probability of occupying states 1 and 2.

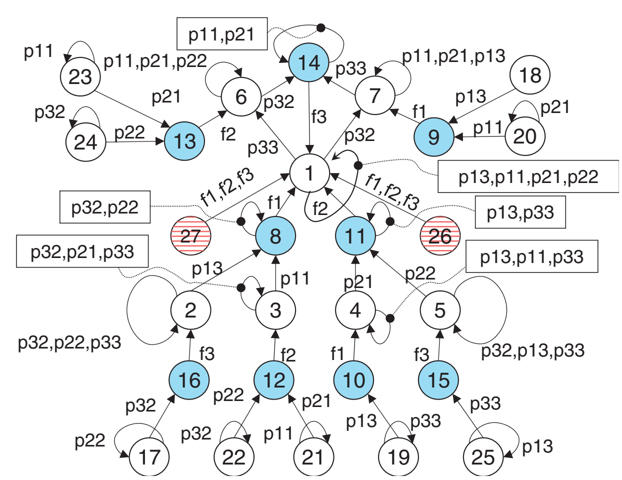


Figure 5. Optimal plant configuration for specification 1.

6. Summary, conclusions, and recommendations for future work

This paper presents the theory, formulation, and validation of optimal supervisory control policies for dynamical systems, modelled as probabilistic finite state automata. The procedure for synthesis of the optimal control policy relies on a (renormalized) signed real measure of regular languages (Chattopadhyay and Ray 2006a) to construct the performance index. The language measure is based on the state transition

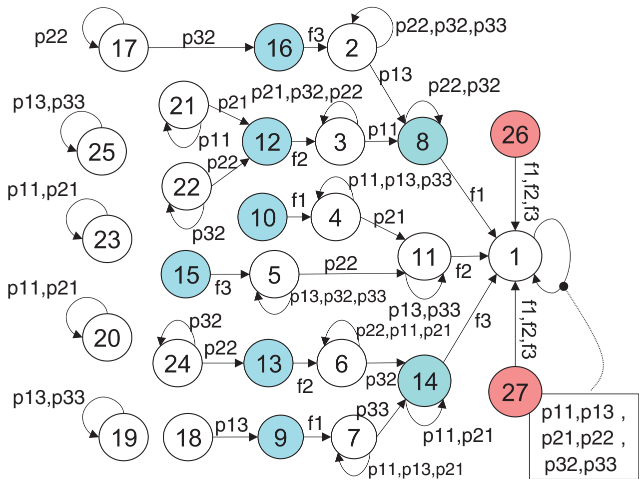


Figure 6. Optimal plant configuration for specification 2.

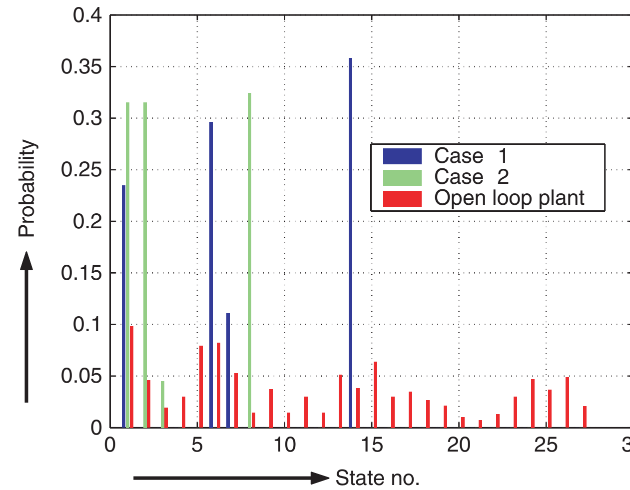


Figure 8. Stable state probability: unsupervised and supervised plants.

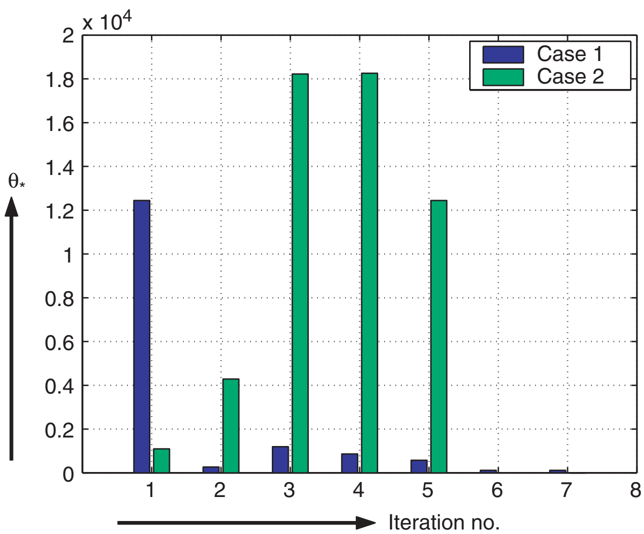


Figure 7. Computed $\theta_\star$ for each iteration.

probability matrix of the underlying finite-state Markov chain model of the process and a characteristic vector of state weights, which serves as the control specification.

The main contribution of the paper is reformulation of the optimal supervisor synthesis algorithm (Ray *et al*. 2004, 2005) for probabilistic finite state plant models in terms of the renormalized measure and extension of the technique to general non-terminating probabilistic models. Specifically, the work reported in this paper removes a fundamental restriction of earlier analysis (Ray *et al*. 2004, Ray 2005), namely, each row sum of the state transition cost matrix $\mathbf{\Pi}$ being strictly less than one, instead of being exactly equal to one. The novel concept of language-based control synthesis, presented in this paper, allows quantification of plant performance instead of solely relying on its qualitative performance (e.g., permissiveness), which is the current state of the art for discrete event supervisory control (Ramadge and Wonham 1987, Cassandras and Lafortune 1999).

The following conclusion is drawn in view of using the language measure for construction of the performance index for deriving an optimal control policy. Like any other optimization procedure, it is possible to choose different performance indices to arrive at different optimal policies for discrete event supervisory control. Nevertheless, usage of the language measure provides a systematic procedure for precise comparative evaluation of different supervisors so that the optimal control policy(ies) can be unambiguously identified. These theoretical results also lay the foundation for extension of the language-measure-theoretic framework to plant modelling and control, where all events may not be observable at the supervisory level.

The paper provides details of the algorithms that are required for synthesis of the optimal supervisory control policy. These algorithms are executable in real time on commercially available platforms. Computational complexity of the presented algorithms is polynomial in the number of plant model states. The concepts are elucidated with simple examples and a relevant engineering example. As such it is straight-forward to develop real-time software codes in standard languages, based on the algorithms provided in this paper.

There are several issues that need to be addressed for implementation of the theory of discrete-event supervisory control in an operating plant. For example, the events must be generated in real time, based on physical measurements, to provide the supervisor with the

current information on the plant; this is beyond what is done off-line for construction of the plant model and control synthesis. Similarly, the event disabling/enabling decisions of the supervisor must be translated in real time as appropriate actions to control the plant.

6.1 *Recommendations for future research*

Synthesis of supervisory control systems may become a significant challenge if some of the events are delayed, intermittent, or not observable at all, possibly due to sensor faults or malfunctions in network communication links. In that case, the control algorithms may turn out to be computationally very complex because of delayed or lost information on the plant dynamics. Future work in this direction should involve research on construction of language measures under partial observation (Chattopadhyay and Ray 2006b) and associated synthesis of optimal control policies to mitigate the detrimental effects of loss of observability. The latter research could be an extension of the work on optimal control under full observation, reported in this paper.

It would be a challenging task to extend the concept of (regular) language measure for languages higher up in the Chomsky hierarchy (Hopcroft *et al.* 2001) such as context-free and context-sensitive languages. This extension would lead to controller synthesis when the plant dynamics is modelled by non-regular languages such as the Petri net (Cassandras and Lafortune 1999, Murata 1989). The research thrust should focus on retaining the polynomial order of computational complexity.

Another critical issue is how to extend the language measure for timed automaton, especially if the events are observed with varying delays at different states. Another research topic that may also be worth investigating is: how to extend the *GF*(2) field, over which the vector space of languages is defined (Ray 2005), to richer fields like the set of real numbers.

Areas of future research also include applications of the language measure in anomaly detection, model identification, model order reduction, and analysis and synthesis of interfaces between the continuously-varying and discrete-event spaces in the language-measure setting. Future research for advancement of the theory of optimal supervisory control for discrete event systems include the following areas:

- Robustness of the control policy relative to unstructured and structured uncertainties in the plant model including variations in the language measure parameters (Lagoa *et al.* 2005)
- Control synthesis under partial observation to accommodate loss of observability at the supervisory level possibly due to sensor faults or communication link failures (Chattopadhyay and Ray 2006b)
- Construction of grammar-based measures, instead of memory-less state-based measures (Chattopadhyay and Ray 2005), for non-regular languages when details of transitions in plant dynamics cannot be captured by finitely many states

Acknowledgement

This work has been supported in part by the U.S. Army Research laboratory and the U.S. Army Research Office under Grant Nos. DAAD19-01-1-0646 and W911NF-06-1-0469.

Appendix A: Derivatives of renormalized measure

This appendix establishes bounds on the derivatives of the renormalized measure $\boldsymbol{\nu}(\theta)$ for all $\theta \in (0,1)$ and computes the limits of the derivatives as $\theta \to 0^+$ as an extension of what was reported in the previous publication (Chattopadhyay and Ray 2006a).

The main result on boundedness of the derivatives of $\nu(\theta)$ are presented as propositions. Specifically, the results reported in Chattopadhyay and Ray (2006a) are combined as the next two propositions.

Proposition A.1: *Let* $\Psi(\theta) \triangleq \theta[\mathbb{I} - (1-\theta)\mathbf{P}]^{-1}$, *where* $\mathbf{P}$ *is a* $(n \times n)$ *stochastic matrix and* $n \in \mathbb{N}$. *Then,*

(i) $\forall\, k \in \mathbb{N}\backslash\{1\}$

$$\lim_{\theta\to 0^+} \frac{\partial^k \Psi(\theta)}{\partial\theta^k} = -k \lim_{\theta\to 0^+} \frac{\partial^k - 1\Psi(\theta)}{\partial\theta^k - 1}[\mathbf{P} + \mathscr{P}][\mathbb{I} - \mathbf{P} + \mathscr{P}]^{-1}$$

(ii) $$\lim_{\theta\to 0^+} \frac{\partial^k \Psi(\theta)}{\partial\theta^k} = \begin{cases} [\mathbb{I} - \mathbf{P} + \mathscr{P}]^{-1} - \mathscr{P}, & \textit{if } k = 1 \\ (-1)^k k! [\mathbb{I} - \mathbf{P} + \mathscr{P}]^{-\mathbf{1}} & \\ \quad \times \left[\mathbb{I} - [\mathbb{I} - \mathbf{P} + \mathscr{P}]^{-1}\right]^{k-1}, & \textit{if } k \in \mathbb{N}\backslash\{1\}. \end{cases}$$

Proof: Given in Chattopadhyay and Ray (2006a, § 3, pp. 1111–1112 as Corollary 3 and Corollary 6). □

The next proposition establishes bounds on the derivatives of $\boldsymbol{\nu}(\theta)$ in an elementwise sense by computing bounds on the induced sup-norm of the derivatives of $\Psi(\theta)$. Recall that $\boldsymbol{\chi}$ has been defined to have infinity norm equal to 1.

Proposition A.2

$$\left\| \frac{\partial^k \nu(\theta)}{\partial \theta^k} \right\|_\infty \le k!\, 2^{k+1} \left(\inf_{\alpha \neq 0} \left\| \left[\mathbb{I} - \mathbf{P} + \alpha \mathscr{P} \right]^{-1} \right\|_\infty \right)^k \quad \forall \theta \in [0,1].$$

Proof: Given in of Chattopadhyay and Ray (2006a, § 3, p. 1113 as Proposition 5). □

Proposition A.3: Denoting the ith element of the kth derivative of the measure vector as $(\partial^k \nu(\theta)/\partial \theta^k)\big|_i$, it follows that

$$\forall k \in \{1, \ldots, n\},\ \frac{\partial^k \nu(\theta)}{\partial \theta^k}\Big|_i = \frac{\partial^k \nu(\theta)}{\partial \theta^k}\Big|_j$$
$$\Longrightarrow \forall \theta \in [0,1],\ \nu(\theta)\big|_i = \nu(\theta)\big|_j,$$

where n is the number of states in the plant model.

Proof: First it is noted that

$$\begin{aligned} \nu(\theta) &= \theta \sum_{k=0}^{\infty} (1-\theta)^k \mathbf{P}^k \boldsymbol{\chi}\ \forall \theta \in (0,1] \\ &= \theta \sum_{k=0}^{\infty} \boldsymbol{\Pi}^k(\theta) \boldsymbol{\chi}\ \forall \theta \in (0,1]. \end{aligned} \tag{26}$$

Since $\boldsymbol{\Pi}(\theta)$ is a matrix of dimension $n \times n$, it follows from the Cayley–Hamilton Theorem (Bapat and Raghavan 1997) that integral powers of $\boldsymbol{\Pi}(\theta)$ can be expressed as polynomials of degree $n-1$ as follows:

$$\forall r \in \mathbb{N},\ \boldsymbol{\Pi}^r(\theta) = \sum_{k=0}^{n-1} c_k \boldsymbol{\Pi}^k(\theta) \quad \text{with } c_k \in \mathbb{C}. \tag{27}$$

Since each term in the summation on the left hand side of equation (26) is a polynomial in θ of degree $n-1$, it follows that the summation is also a polynomial in degree $n-1$ (since the summation exists due to the sub-stochastic property of $\boldsymbol{\Pi}(\theta)$). Then it follows that each element of $\nu(\theta)$ is a polynomial of degree n. The result then follows from continuity. □

Proposition A.4: For any stochastic matrix $\mathbf{P}$ of dimension $n \times n$, the complexity of computing the limiting matrix $\mathscr{P}$ is of the order of $O(n^3)$.

Proof: Since the limit $\lim_{k\to\infty} (1/k) \sum_{j=0}^{k-1} \mathbf{P}^j = \mathscr{P}$ always exists, it is possible to compute $\mathscr{P}$ within any specified precision simply by computing the sum $\sum_{j=0}^{k-1} \mathbf{P}^j$ followed by division by k, for a large enough value of k. The procedure is summarized in Algorithm 6.

Algorithm 6: Computation of $\mathscr{P}$

input : P, Desired precision eps
output: $\mathscr{P}$

1 **begin**
2   Set $Q^{[0]} = \mathbf{0}$;   /* Zero Matrix */
3   Set $\mathbf{A} = \mathbb{I}$;   /* Indentity Matrix */
4   Set $k = 1$;
5   Set Terminate = **false** ;
6   **while** (Terminate == **false**) **do**
7         $Q^{[k]} = Q^{[k-1]} + \frac{1}{k}\left(\mathbf{A} - Q^{[k-1]}\right)$;
8   **if** $\|Q^{[k]} - Q^{[k-1]}\| <$ eps **then**
9   Terminate == **true** ;
10   **else**
11   k=k+1;
12            $\mathbf{A} = \mathbf{AP}$;
13   **endif**
14   **endw**
15    $\mathscr{P} = Q^{[k]}$;
16 **end**

Referring to Line 7 of Algorithm 6, it is observed that $Q^{[k]}$ is a stochastic matrix for all k and hence it follows that the algorithm is guaranteed to terminate in (1/eps) iterations, independent of n. Each iteration involves a single matrix multiplication ($\mathbf{P} \times \mathbf{A}$) and hence algorithmic complexity is of the same order as multiplication of two $n \times n$ matrices, i.e., $\le O(n^3)$.

References

R.B. Bapat and T.E.S. Raghavan, *Nonnegative matrices and Applications*, Cambridge, UK: Cambridge University Press, 1997.

C.G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*, Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006. ISBN 0387333320.

I. Chattopadhyay and A. Ray, "A complex measure for linear grammars, *Demonstratio Mathematica*, 38:761–775, 2005.

I. Chattopadhyay and A. Ray, "Renormalized measure of regular languages", *Int. J. Control*, 79, pp. 1107–1117, 2006a.

I. Chattopadhyay and A. Ray, "A language measure for partially observed discrete event systems", *Int. J. Control*, 79, pp. 1074–1086, 2006b.

S.L. Chung, S. Lafortune, and F. Lin, "Supervisory control using variable look-ahead policies", *Discrete Event Dynamic Systems: Theory and Applications*, 4, pp. 237–268, July 1994.

E-E. Doberkat, *Stochastic Automata: Stability, Nondeterminism, and Prediction*, volume 113 of *Lecture Notes in Computer Science*, Springer, Secaucus, NJ, USA, 1981. ISBN 3-540-10835-1.

V.K. Garg, R. Kumar, and S.I. Marcus, "A probabilistic language formalism for stochastic discrete event systems", *IEEE Transanctions on Automatic Control*, 44, pp. 280–293, 1999.

V.K. Garg, "An algebraic approach to modeling probabilistic discrete event systems", *Proceedings of 1992 IEEE Conference on Decision and Control*, pp. 2348–2353, Tucson, AZ, December 1992.

V.K. Garg, "Probabilistic languages for modeling of DEDs", in *Proceedings of 1992 IEEE Conference on Information and Sciences*, pp. 198–203, Princeton, NJ, March 1992.

J.E. Hopcroft, R. Motwani and J.D. Ullman, *Introduction to Automata Theory, Languages, and Computation, 2nd ed.*, Massachusetts, USA: Addison-Wesley, 2001.

R. Kumar and V.K. Garg, "Control of stochastic discrete event systems modeled by probabilistic languages", *IEEE Transactions on Automatic Control*, 46, pp. 593–606, 2001. URL citeseer.ist.psu.edu/278517.html.

C.M. Lagoa, J. Fu, and A. Ray, "Robust optimal control of regular languages", *Automatica*, 41, pp. 1339–1345, 2005.

M. Lawford and W.M. Wonham, "Supervisory control of probabilistic discrete event systems", in *Proceedings of 36th Midwest Symposium on Circuits and Systems, Detroit, Michigan, USA*, pp. 327–331, 1993.

E.T. Lee and L.A. Zadeh, "Note on fuzzy languages", *Information Sciences*, 1, pp. 421–434, 1969.

R. Milner, *Communication and concurrency*, Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1989. ISBN 0-13-115007-3.

M.K. Molloy, "Performance analysis using stochastic petri nets", *IEEE Transactions on Computers*, C-31, pp. 913–917, September 1982.

H. Mortzavian, "Controlled stochastic languages", in *Proceedings of 1993 Allerton Conference*, pp. 938–947, Urbana, IL, 1993.

T. Murata, "Petri nets: Properties, analysis and applications", in *Proceedings IEEE*, 77, pp. 541–540, April 1989.

A. Paz, *Introduction to probabilistic automata (Computer science and applied mathematics)*. Orlando, FL, USA: Academic Press, Inc. 1971. ISBN 0125476507.

S. Phoha, E. Peluso, and R.L. Culver, "A high fidelity ocean sampling mobile network (samon) simulator", *IEEE Journal of Oceanic Engineering, Special Issue on Autonomous Ocean Sampling Networks*, 26, pp. 646–653, 2002.

M.O. Rabin, "Probablistic automata", *Information and Control*, 6, pp. 230–245, 1963.

P.J. Ramadge and W.M. Wonham, "Supervisory control of a class of discrete event processes", *SIAM J. Control and Optimization*, 25, pp. 206–230, 1987.

A. Ray, J. Fu, and C.M. Lagoa, "Optimal supervisory control of finite state automata", *Int. J. Control*, 77, pp. 1083–1100, 2004.

A. Ray, V. Phoha, and S. Phoha, *Quantitative measure for discrete event supervisory control*.Springer, New York, 2005.

A. Ray, "Signed real measure of regular languages for discrete-event supervisory control, *Int. J. Control*, 78, pp. 949–967, 2005.

R. Sengupta and S. Lafortune. An optimal control theory for discrete event systems, *SIAM Journal on Control and Optimization*, 36, pp. 488–541, 1998. URL citeseer.ist.psu.edu/sengupta98optimal.html.

International Journal of Control
Vol. 80, No. 5, May 2007, 789–799

Taylor & Francis
Taylor & Francis Group

Generalized language measure families of probabilistic finite state systems

I. CHATTOPADHYAY and A. RAY*

Mechanical Engineering Department, Pennsylvania State University, University Park, PA 16802, USA

(*Received 11 August 2006; in final form 28 December 2006*)

The signed real measure of regular languages has been introduced and validated in recent literature for quantitative analysis of discrete-event systems. This paper reports generalizations of the language measure, which can serve as performance indices for synthesis of optimal discrete-event supervisory decision and control laws. These generalizations eliminate a user-selectable parameter in the original concept of language measure. The concepts are illustrated with simple examples.

1. Introduction

In the discrete-event setting, a finite-state automaton (FSA) model of a physical plant is a generator of its regular language, whose behaviour is constrained by the supervisor (or controller) to meet a given specification. A signed real measure of regular languages has been reported in Ray (2005) and Ray *et al.* (2005) to provide a mathematical framework for quantitative comparison of controlled sublanguages. In this work, each transition is assigned a cost, similar to its probability measure that can be quantitatively evaluated from physical experimentation or extensive simulation on a test bed. Each state of the FSA model is assigned a signed real weight whose upper and lower bounds are normalized to 1 and -1, respectively. The measure of a given event trace is obtained as the product of the cost of transitions and the (normalized) weight of the terminating state. The sum of the measures of all traces yield the language measure.

Optimal control of finite state automata has been recently reported Ray *et al.* (2004, 2005) based on the total ordering induced by the language measure as augmentation to the supervisory control theory of Ramadge and Wonham (1987). This work consolidates the theory and applications of optimal supervisory control of regular languages, where the performance index is obtained by combining a real signed measure of the supervised plant language with the cost of disabled event(s). Starting with the (regular) language of an unsupervised plant automaton, the optimal control policy makes a trade-off between the measure of the supervised sublanguage and the associated event disabling cost to achieve the best performance. Like any other optimization procedure, it is possible to choose different performance indices to arrive at different optimal policies for discrete event supervisory control. It is recognized that optimal control of discrete-event systems can be achieved with a cost function that may not qualify as a measure (e.g., Sengupta and Lafortune (1998)). Nevertheless, usage of a language measure as the cost function facilitates precise comparative evaluation of different supervisors so that the appropriate control policy(ies) can be conclusively identified.

From the above perspectives, this paper presents generalizations of the language measure (Ray 2005), each generalization being a formal measure in its own right and having physical implications that are relevant to synthesis of discrete-event supervisory control policies. These generalizations are achieved through a new concept of trace measure that is characterized by both initiating and terminating states as well as the length of the trace and the choice of a vector

*Corresponding author. Eamil: axr2@psu.edu

International Journal of Control
ISSN 0020–7179 print/ISSN 1366–5820 online © 2007 Taylor & Francis
http://www.tandf.co.uk/journals
DOI: 10.1080/00207170601188784

norm (Naylor and Sell 1982). The concept of generalization can be viewed as renormalization (Chattopadhyay and Ray 2006) of the (normalized) language measure (Ray 2005).

The paper is organized in six sections including the present one. Section 2 briefly reviews background concepts on language measure. Section 3 derives measures related to the stationary state probability vector of the finite-state automaton. Section 4 introduces the notion of shaped measures, which allows assignment of selective length-based importance to different traces in the generated language. It is further shown that measures introduced in § 3 can be obtained as limits of sequences of shaped measures. Section 5 presents an example of optimal automaton configurations. Section 6 concludes the paper along with recommendations for future research.

2. Brief review of language measure

This section briefly reviews the concept of signed real measure of regular languages Ray (2005) and Ray *et al.* (2005). Let $G_i \equiv \langle Q, \Sigma, \delta, q_i, Q_m \rangle$ be a trim (i.e., accessible and co-accessible) deterministic finite-state automaton (DFSA) model (Ramadge and Wonham 1987) that represents the discrete-event dynamics of a physical plant, where $Q = \{q_k : k \in I_Q\}$ is the set of states and $I_Q \equiv \{1, 2, \ldots, n\}$ is the index set of states; the automaton starts with the initial state q_i; the alphabet of events is $\Sigma = \{\sigma_k : k \in I_\Sigma\}$, and $I_\Sigma \equiv \{1, 2, \ldots, \ell\}$ is the index set of events; δ: $Q \times \Sigma \to Q$ is the (possibly partial) function of state transitions; and $Q_m \equiv \{q_{m_1}, q_{m_2}, \ldots, q_{m_r}\} \subseteq Q$ is the set of marked (i.e., accepted) states with $q_{m_k} = q_j$ for some $j \in I_Q$.

Let $\Sigma^\star$ be the Kleene closure of Σ, i.e., the set of all finite-length traces made of the events belonging to Σ as well as the empty trace ϵ that is viewed as the identity of the monoid $\Sigma^\star$ under the operation of trace concatenation, i.e., $\epsilon s = s = s\epsilon$. The extension $\delta^\star$: $Q \times \Sigma^\star \to Q$ is defined recursively in the usual sense (Hppcroft *et al.* 2001).

Definition 1: The language $L(G_i)$ generated by a *DFSA* G initialized at the state $q_i \in Q$ is defined as $L(G_i) = \{s \in \Sigma^\star \mid \delta^\star(q_i, s) \in Q\}$.

Definition 2: The language $L_m(G_i)$ marked by a *DFSA* G_i initialized at the state $q_i \in Q$ is defined as $L_m(G_i) = \{s \in \Sigma^\star \mid \delta^\star(q_i, s) \in Q_m\}$.

The language $L(G_i)$ is partitioned into non-marked and marked languages, $L^o(G_i) \equiv L(G_i) - L_m(G_i)$ and $L_m(G_i)$, consisting of event traces that, starting from $q_i \in Q$, terminate at one of the non-marked states in $Q - Q_m$ and one of the marked states in Q_m, respectively. The set Q_m is further partitioned into Q_m^+ and Q_m^-, where Q_m^+ contains all good marked states that are desired to be terminated on and Q_m^- contains all bad marked states that one may not want to terminate on, although it may not always be possible to avoid the bad states while attempting to reach the good states. Accordingly, the marked language $L_m(G_i)$ is further partitioned into $L_m^+(G_i)$ and $L_m^-(G_i)$ consisting of good and bad traces that, starting from q_i, terminate on Q_m^+ and Q_m^-, respectively. Thus, the language $L(G_i)$ is decomposed into null, i.e., $L^o(G_i)$, positive, i.e., $L_m^+(G_i)$, and negative, i.e., $L_m^-(G_i)$ sublanguages. A signed real measure μ: $2^{L(G_i)} \to \mathbb{R} \equiv (-\infty, \infty)$ is constructed for quantitative evaluation of every event trace $s \in L(G_i)$.

Definition 3: The language of all traces that, starting at a state $q_i \in Q$, terminates on a state $q_j \in Q$, is denoted as $L(q_i, q_j)$. That is, $L(q_i, q_j) \equiv \{s \in L(G_i) : \delta^\star(q_i, s) = q_j\}$.

Definition 4: The terminating characteristic function that assigns a normalized signed real weight to state-partitioned sublanguages $L(q_i, q_j)$, $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, n$ is defined as χ: $Q \to [-1,\ 1]$ such that

$$\chi_j \in \begin{cases} [-1,\ 0) & \text{if } q_j \in Q_m^- \\ \{0\} & \text{if } q_j \notin Q_m \\ (0,\ 1] & \text{if } q_j \in Q_m^+. \end{cases} \tag{1}$$

Definition 5: The event cost is conditioned on a *DFSA* state at which the event is generated, and is defined as $\tilde{\pi}: L(G_i)) \times Q \to [0, 1]$ such that $\forall q_j \in Q, \forall \sigma_k \in \Sigma$, $\forall s \in L(G_i))$

(1) $\tilde{\pi}[\sigma_k, q_j] \equiv \tilde{\pi}_{jk} \in [0, 1)$; $\sum_k \tilde{\pi}_{jk} < 1$;
(2) $\tilde{\pi}[\sigma, q_j] = 0$ if $\delta(q_j, \sigma)$ is undefined; $\tilde{\pi}[\epsilon, q_j] = 1$;
(3) $\tilde{\pi}[\sigma_k s, q_j] = \tilde{\pi}[\sigma_k, q_j]\ \tilde{\pi}[s, \delta(q_j, \sigma_k)]$.

The event cost matrix is defined as $\widetilde{\mathbf{\Pi}}_{ij} = \tilde{\pi}_{ij}$ with $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, m\}$ where the automaton has n states and cardinality of the event alphabet Σ is m.

An application of the induction principle to part (3) in Definition 5 shows $\tilde{\pi}[st, q_j] = \tilde{\pi}[s, q_j]\tilde{\pi}[t, \delta^\star(q_j, s)]$. The condition $\sum_k \tilde{\pi}_{jk} < 1$ provides a sufficient condition for the existence of the real signed measure (Ray 2005). Next a measure of sublanguages of the plant language $L(G_i)$ is formulated in terms of the signed characteristic function χ and the non-negative event cost $\tilde{\pi}$.

Definition 6: The state transition cost, π: $Q \times Q \to [0, 1)$, of the *DFSA* G_i is defined as follows: $\forall q_i, q_j \in Q$,

$$\pi_{ij} = \begin{cases} \sum_{\sigma \in \Sigma} \tilde{\pi}[\sigma, q_i], & \text{if } \delta(q_i, \sigma) = q_j \\ 0 & \text{if} \{\delta(q_i, \sigma) = q_j\} = \emptyset. \end{cases} \tag{2}$$

Consequently, the $n \times n$ state transition cost $\mathbf{\Pi}$-matrix is defined as $\mathbf{\Pi}_{ij} = \pi_{ij}$ with $i, j \in \{1, \dots, n\}$ where the number of states in the automaton is n.

Although the preceding analysis reported in Ray (2005) and Ray *et al.* (2005) was intended for non-probabilistic regular languages, the event costs can be interpreted as conditional probabilities of event occurrence. A brief discussion on the physical interpretation of the event costs is given in Ray (2005) to explain this issue. Furthermore, an element π_{jk} of the Π-matrix is conceptually similar to the state transition probability of a Markov chain or a semi-Markov chain with the exception that the equality condition $\sum_k \pi_{jk} = 1$ is not satisfied. Specifically, the inequality $\sum_k \pi_{jk} < 1, \; j = 1, 2, \dots, n$ provides a sufficient condition for the language measure to be finite. This implies that the preceding analysis is applicable to the case of terminating probabilistic languages (Garg 1992a, b) that have a non-zero probability of termination (arising from either intentional design or unmodelled dynamics of the plant automaton) at each state. If the probability of termination at each state, or equivalently the probability of transition to the (deadlock) dump state from each of the other states $q_i \in Q$, is set identically equal to $\theta \in (0, 1)$, then the $\widetilde{\mathbf{\Pi}}$-matrix and the $\mathbf{\Pi}$-matrix can be θ-parameterized as follows (Chattopadhyay and Ray 2006):

$$\widetilde{\mathbf{\Pi}}(\theta) \equiv (1-\theta)\widetilde{\mathbf{P}} \quad \text{and} \quad \mathbf{\Pi}(\theta) \equiv (1-\theta)\mathbf{P}, \tag{3}$$

where $\widetilde{\mathbf{P}}$ is the event matrix (also known as the morph matrix), which is derived from experimental data or simulation data (Ray 2005) and the resulting stochastic state transition matrix $\mathbf{P}$ is obtained from $\widetilde{\mathbf{P}}$ in a way similar to equation (2). Since $\mathbf{P}$ is a stochastic matrix (i.e., $\sum_j \mathbf{P}_{ij} = 1 \; \forall i \in \{1, \dots, n\}$), the row sums $\sum_j \pi_{ij} = (1-\theta) < 1, \; j = 1, 2, \dots, n$ (see Definition 6) make $\mathbf{\Pi}$ a contraction operator with the magnitude of each of its eigenvalues being less than or equal to $(1-\theta)$; consequently, $[\mathbb{I} - \mathbf{\Pi}]$ becomes invertible (Ray 2005).

In the sequel, the preceding measure construction is generalized and the notion of language measure is extended to non-terminating models by first assuming a uniform non-zero probability of termination θ at each state and then computing the limit as $\theta \to 0^+$, i.e., the probability of termination approaching zero. The resulting θ-parameterized model coincides with the desired non-terminating model in the limit (Chattopadhyay and Ray 2006).

Definition 7: The θ-parameterized measure of the language $L(q_i, q_j)$ is defined in terms of its traces (see Definitions 3, 4 and 5) as

$$\mu^\theta(\{s\}) \equiv \tilde{\pi}(s, q_i)\chi_j, \quad \forall s \in L(q_i, q_j) \tag{4}$$

$$\mu^\theta\big(L(q_i, q_j)\big) \equiv \sum_{s \in L(q_i, q_j)} \mu^\theta(\{s\}). \tag{5}$$

Then, the measure of the language $L(G_i)$ of a *DFSA* G_i, initialized at the state $q_i \in Q$, is defined as

$$\mu^\theta(L(G_i)) = \sum\nolimits_j \mu^\theta\big(L(q_i, q_j)\big) \tag{6}$$

It is shown in Ray (2005) that the measure $\mu_i^\theta \equiv \mu^\theta(L(G_i))$ can be expressed as: $\mu_i^\theta = \sum_j \pi_{ij} \mu_j^\theta + \chi_i$. In vector notation, the θ-parameterized language measure vector is expressed by making use of equation (3) as

$$\boldsymbol{\mu}^\theta = [\mathbb{I} - (1-\theta)\mathbf{P}]^{-1}\boldsymbol{\chi}, \tag{7}$$

where the measure vector $\boldsymbol{\mu}^\theta \equiv [\mu_1^\theta \; \mu_2^\theta \; \cdots \; \mu_n^\theta]^T$ and the terminating characteristic vector $\boldsymbol{\chi} \equiv [\chi_1 \; \chi_2 \; \cdots \; \chi_n]^T$. Note that $\lim_{\theta \to 0^+} \boldsymbol{\mu}^\theta$ of the normalized language measure does not exist. This problem has been circumvented via renormalization (Chattopadhyay and Ray 2006) as explained below.

The regular language $L(G_i)$ is a sublanguage of the Kleene closure $\Sigma^\star$ of the alphabet Σ, for which the automaton states can be merged into a single state. Then, $\mathbf{P}$ degenerates to the 1×1 identity matrix and the terminating characteristic vector $\boldsymbol{\chi}$ becomes one-dimensional and can be assigned as $\boldsymbol{\chi} = 1$ by normalization. Consequently, the measure vector $\boldsymbol{\mu}^\theta$ in equation (7) degenerates to a scalar measure θ^{-1}. The renormalized measure is obtained from equation (7) after normalization with respect to θ^{-1}.

$$\boldsymbol{\vartheta}^\theta|_1 = \theta \, [\mathbb{I} - (1-\theta)\mathbf{P}]^{-1}\boldsymbol{\chi}. \tag{8}$$

3. Generalization of language measure

This section generalizes the notion of language measure μ^θ (see Definition 7 and equation (7)), which also leads to a renormalized measure ϑ^θ (see equation (8)). This is achieved by redefining the measure of individual traces in terms of an initiating characteristic function ξ: $Q \longmapsto [0, 1]$ that assigns a positive weight to each initiating state q_i and serves as a renormalizing factor (i.e., a multiplicative constant) for the measure of the traces initiating from the respective state. Figure 1 illustrates the relationship among the initiating and terminating characteristics. Different initiating

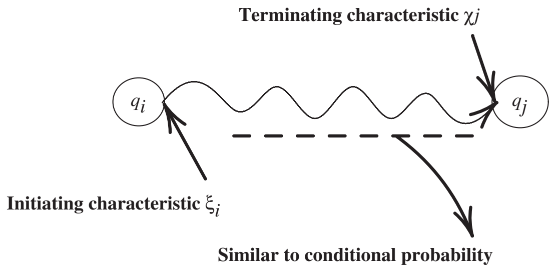


Figure 1. Generalization of langauge measure.

characteristics lead to different renormalized language measures that may have different physical interpretations.

Definition 8: The θ-parameterized generalized measure of a singleton event trace set $\{s\} \subseteq L(q_i, q_j) \subseteq L(G_i)$ in the σ-algebra $2^{L(G_i)}$ is defined as

$$\vartheta^\theta(\{s\}) \equiv \xi_i \mu^\theta(\{s\}) = \xi(q_i)\tilde{\pi}(s, q_i)\chi_j, \quad \forall s \in L(q_i, q_j). \quad (9)$$

The generalized measure of $L(q_i, q_j)$ is defined as

$$\vartheta^\theta\big(L(q_i, q_j)\big) \equiv \sum_{s \in L(q_i, q_j)} \vartheta^\theta(\{s\}). \quad (10)$$

The generalized measure of a *DFSA* G_i, initialized at the state $q_i \in Q$, is denoted as $\vartheta_i^\theta \equiv \vartheta^\theta(L(G_i)) = \sum_j \vartheta^\theta(L(q_i, q_j))$.

Now it is ascertained that Definition 8 satisfies the properties of a measure on the defined σ-algebra.

Proposition 1: *The generalized measure* $\vartheta^\theta: 2^{L(G_i)} \longmapsto \mathbb{R}$ *is defined on the measure space* $(L(G_i), 2^{L(G_i)}, \vartheta^\theta)$.

Proof: It suffices to establish σ-additivity from the following fact. For a fixed $\theta \in (0, 1)$, ϑ_i^θ is the product of ξ_i (which is a constant) and μ_i^θ which is a signed real measure on the σ-algebra $2^{L(G_i)}$. □

A special family of initiating characteristic functions is considered for the generalized language measure.

Definition 9: Let $\Psi(\theta) \equiv [\mathbb{I} - (1-\theta)\mathbf{P}]^{-1}$. The ℓ_p-family of initiating characteristic functions is defined as

$$\xi_i^p(\theta) = \big|\big|\Psi(\theta)_{i\circ}\big|\big|_p^{-1} \quad \forall p \in [1, \infty], \quad \forall \theta \in (0, 1), \quad (11)$$

where $\| \bullet \|_p$ denotes the ℓ_p-norm of $\bullet$; and ith row of a matrix $\mathbf{M}$ is denoted as $\mathbf{M}_{i\circ}$ and the jth column as $\mathbf{M}_{\circ j}$.

Remark 1: Note that $\lim_{\theta \to 0^+} \xi_i^p(\theta)$ does not exist due to non-invertibility of the operator $[\mathbb{I} - \mathbf{P}]$. However, $(\theta^{-1}\xi_i^p(\theta))|_{\theta=0}$ is well-defined by virtue of norm continuity (Naylor and Sell 1982) and hence $\lim_{\theta \to 0^+} \theta^{-1}\xi_i^p(\theta)$ exists.

Lemma 1: *For $p = 1$, the initiating characteristic*

$$\xi_i^1(\theta) = \theta, \quad \forall i \forall \theta \in (0, 1]. \quad (12)$$

Proof: Let $\boldsymbol{e} = [1 \cdots 1]^T$. Since $\mathbf{P}$ is stochastic, non-negativity of $\Psi(\theta)$ follows from the following expansion:

$$\Psi(\theta) = \sum_{k=0}^{\infty} (1-\theta)^k \mathbf{P}^k \quad \forall \theta \in (0, 1]$$
$$\Longrightarrow \Psi(\theta)\boldsymbol{e} = \sum_{k=0}^{\infty} (1-\theta)^k \mathbf{P}^k \boldsymbol{e} = \sum_{k=0}^{\infty} (1-\theta)^k \boldsymbol{e} = \theta^{-1}\boldsymbol{e}$$

which implies that $\| \Psi(\theta)_{i\circ} \|_1 = \theta^{-1} \ \forall i \Longrightarrow \xi_i^1 = \theta$. □

The θ-parameterized generalized measure for $p = 1$ is obtained in the vector notation as

$$\boldsymbol{\vartheta}^\theta|_1 = \theta[\mathbb{I} - (1-\theta)\mathbf{P}]^{-1}\boldsymbol{\chi} \quad (13)$$

which is identical to the renormalized measure in equation (8).

In general, the θ-parameterized generalized measure for $p \in [1, \infty]$ is obtained in the matrix notation as

$$\boldsymbol{\vartheta}^\theta|_p = \begin{bmatrix} \xi_1^p(\theta) & \cdots & 0 \\ \vdots & \xi_i^p(\theta) & \vdots \\ 0 & \cdots & \xi_n^p(\theta) \end{bmatrix} [\mathbb{I} - (1-\theta)\mathbf{P}]^{-1}\boldsymbol{\chi}. \quad (14)$$

Non-negativity of $\mathbf{P}$ and invertibility of $[I - (1-\theta)\mathbf{P}]$ guarantee that $\|\Psi(\theta)_{i\circ}\|_p \in (0, \infty) \ \forall i$, which implies $\xi_i^p(\theta) \in (0, \infty) \ \forall \ p \in [1, \infty] \ \forall \ \theta \in (0, \infty)$.

3.1 *Limiting values of* $\boldsymbol{\vartheta}^\theta|_p$ *as* $\theta \to 0^+$

This section computes the generalized measures $\boldsymbol{\vartheta}^\theta|_p$ as $\theta \to 0^+$, based on the state transition probability matrix $\mathbf{P}$ of a stationary Markov chain with finitely many states. Then, $\mathbf{P}$ is a stochastic matrix. That is, $\mathbf{P}$ is non-negative with each row sum being identically equal to unity (Bapat and Raghavan 1997).

Proposition 2: *For every stochastic matrix* $\mathbf{P}$, *the following limit exists*

$$\lim_{k \to \infty} \frac{1}{k} \sum_{j=0}^{k-1} \mathbf{P}^j = \mathscr{P}, \quad (15)$$

where $\mathscr{P}$ is a stochastic matrix. Furthermore, $\mathscr{P}$ commutes with $\mathbf{P}$ and is idempotent. That is,

$$\mathbf{P}\mathscr{P} = \mathscr{P}\mathbf{P} = \mathscr{P} = \mathscr{P}^2. \quad (16)$$

Proof: The proof is given in Bapat and Raghavan (1997). $\square$

Since $\mathbf{P}$ is a stochastic matrix, $[\mathbb{I}-\mathbf{P}]e=0$ where $e\equiv[1,1,\ldots,1]^T$. Therefore, $[\mathbb{I}-\mathbf{P}]$ is not invertible for any stochastic matrix $\mathbf{P}$; however, $[\mathbb{I}-(1-\theta)\mathbf{P}]$ is always invertible for $\theta\in(0,1)$. The lemma to the next proposition shows that $[\mathbb{I}-\mathbf{P}+\mathscr{P}]$ is invertible.

Proposition 3: *The matrix* $[\mathbb{I}-\mathbf{P}+\alpha\mathscr{P}]$ *is invertible for all* $\alpha\neq 0$.

Proof: The proof is based on the commutative and idempotent properties of $\mathscr{P}$ in equation (16) and uses the principle of contradiction.

Let $[\mathbb{I}-\mathbf{P}+\alpha\mathscr{P}]$ be non-invertible for an arbitrary $\alpha\neq 0$. Then, there is a vector $\vartheta\neq 0$ such that

$$\begin{aligned}&[\mathbb{I}-\mathbf{P}+\alpha\mathscr{P}]\vartheta=0\\&\Rightarrow[\mathbf{P}-\alpha\mathscr{P}]\vartheta=\vartheta\Rightarrow\alpha\mathscr{P}[\mathbf{P}-\alpha\mathscr{P}]\vartheta=\alpha\mathscr{P}\vartheta\\&\Rightarrow[\alpha\mathscr{P}-\alpha^2\mathscr{P}]\vartheta=\alpha\mathscr{P}\vartheta\\&\Rightarrow\alpha^2\mathscr{P}\vartheta=0\Rightarrow\mathscr{P}\vartheta=0\text{ because }\alpha\neq 0.\end{aligned}$$

Hence, $\mathbf{P}\vartheta=\mathbf{P}\vartheta-\alpha\mathscr{P}\vartheta=[\mathbf{P}-\alpha\mathscr{P}]\vartheta=\vartheta$

$$\Rightarrow\mathbf{P}^k\vartheta=\vartheta\quad\forall k\in\mathbb{N}\cup\{0\},$$

which implies

$$\left(\frac{1}{k}\sum_{j=0}^{k-1}P^j\right)\vartheta=\vartheta\ \forall k\Longrightarrow\lim_{k\to\infty}\left(\frac{1}{k}\sum_{j=0}^{k-1}P^j\right)\vartheta=\mathscr{P}\vartheta=\vartheta$$
$$\Longrightarrow\vartheta=0\quad\text{because }\mathscr{P}\vartheta=0.$$

This is a contradiction. $\square$

Lemma 2: *The matrix* $[\mathbb{I}-\mathbf{P}+\mathscr{P}]$ *is invertible.*

Proof: The proof follows by setting $\alpha=1$ in Proposition 3. $\square$

Proposition 4:

$$[\mathbf{P}-\alpha\mathscr{P}]^k=\mathbf{P}^k-\left[1-(1-\alpha)^k\right]\mathscr{P},\quad\forall k\in\mathbb{N}\ \forall\alpha\neq 0.\tag{17}$$

Proof: The above identity is valid for $k=0$ and $k=1$. It is also true for $k=2$ by virtue of the commutative and idempotent properties of $\mathscr{P}$ in equation (16). The proof follows directly by the method of induction. $\square$

Lemma 3: $[\mathbf{P}-\mathscr{P}]^k=\mathbf{P}^k-\mathscr{P}\ \ \forall k\in\mathbb{N}$.

Proof: The proof follows by setting $\alpha=1$ in Proposition 4. $\square$

Proposition 5:

$$\lim_{\theta\to 0^+}\theta[\mathbb{I}-(1-\theta)\mathbf{P}]^{-1}=\mathscr{P}.\tag{18}$$

Proof: For $\theta\in(0,1)$, it follows from equation (16) and Lemma 3 that

$$\begin{aligned}&\theta[\mathbb{I}-(1-\theta)\mathbf{P}]^{-1}-\mathscr{P}\\&\quad=\theta\sum_{k=0}^{\infty}\left((1-\theta)^k\mathbf{P}^k\right)-\theta\sum_{k=0}^{\infty}(1-\theta)^k\ \mathscr{P}\\&\quad=\theta\sum_{k=0}^{\infty}(1-\theta)^k\left(\mathbf{P}^k-\mathscr{P}\right)\\&\quad=\theta\sum_{k=0}^{\infty}(1-\theta)^k\left(\mathbf{P}-\mathscr{P}\right)^k\text{ by Lemma 3}\\&\quad=\theta[\mathbb{I}-(1-\theta)(\mathbf{P}-\mathscr{P})]^{-1}\text{ by Lemma 2}\\&\quad\Rightarrow\lim_{\theta\to 0^+}\left(\theta[\mathbb{I}-(1-\theta)\mathbf{P}]^{-1}-\mathscr{P}\right)\\&\quad=\lim_{\theta\to 0^+}\theta[\mathbb{I}-(1-\theta)(\mathbf{P}-\mathscr{P})]^{-1}.\end{aligned}$$

Since, for continuous functions $f(\cdot)$ and $g(\cdot)$ with

$$\begin{aligned}&\lim_{\theta\to 0^+}f(\theta)=0\quad\text{and}\quad\lim_{\theta\to 0^+}g(\theta)<\infty\\&\Longrightarrow\lim_{\theta\to 0^+}f(\theta)g(\theta)=0,\end{aligned}$$

it follows from Lemma 2 that

$$\begin{aligned}&\lim_{\theta\to 0^+}\theta[\mathbb{I}-(1-\theta)(\mathbf{P}-\mathscr{P})]^{-1}=0\\&\quad\Rightarrow\lim_{\theta\to 0^+}\left(\theta[\mathbb{I}-(1-\theta)\mathbf{P}]^{-1}-\mathscr{P}\right)=0.\end{aligned}$$

The proof is thus complete. $\square$

Proposition 6: *For every stochastic matrix* $\mathbf{P}$, *the generalized measure is expressed as*

$$\boldsymbol{\vartheta}^0|_p\equiv\lim_{\theta\to 0^+}\boldsymbol{\vartheta}^\theta|_p=\begin{Bmatrix}\vdots\\\dfrac{\mathscr{P}_{i\circ}\boldsymbol{\chi}}{\|\mathscr{P}_{i\circ}\|_p}\\\vdots\end{Bmatrix},\tag{19}$$

where $\mathscr{P}_{i\circ}$ *is the ith row of* $\mathscr{P}$.

Proof: Following equation (11) in Definition 9, it suffices to show that

$$\lim_{\theta\to 0}\begin{bmatrix}\xi_1^p(\theta) & \cdots & 0\\ \vdots & \xi_i^p(\theta) & \vdots\\ 0 & \cdots & \xi_n^p(\theta)\end{bmatrix}[\mathbb{I}-(1-\theta)\mathbf{P}]^{-1} = \begin{bmatrix}\|\mathscr{P}_{1\circ}\|_p & \cdots & 0\\ \vdots & \|\mathscr{P}_{i\circ}\|_p & \vdots\\ 0 & \cdots & \|\mathscr{P}_{n\circ}\|_p\end{bmatrix}^{-1}\begin{bmatrix}\mathscr{P}_{1\circ}\\ \cdots\\ \mathscr{P}_{n\circ}\end{bmatrix}$$

The above identity is a direct consequence of the following two relations:

$$\lim_{\theta\to 0}\theta\big[\mathbb{I}-(1-\theta)\mathbf{P}\big]^{-1} = \mathscr{P}$$

$$\lim_{\theta\to 0^+}\theta^{-1}\begin{bmatrix}\xi_1^p(\theta) & \cdots & 0\\ \vdots & \xi_i^p(\theta) & \vdots\\ 0 & \cdots & \xi_n^p(\theta)\end{bmatrix} = \begin{bmatrix}\|\mathscr{P}_{1\circ}\|_p & \cdots & 0\\ \vdots & \|\mathscr{P}_{i\circ}\|_p & \vdots\\ 0 & \cdots & \|\mathscr{P}_{n\circ}\|_p\end{bmatrix}^{-1}.$$

The first relation is a restatement of equation (18) in Proposition 6. The second relation is obtained from continuity of norm in equation (11) (see Remark 1). □

We now consider the special class of primitive (i.e., irreducible and acyclic (Bapat and Raghavan 1997)) stochastic matrices. The restriction of primitivity is valid for many applications such as finite-state machines without any deadlock or local livelock. A primitive stochastic matrix **P** has the following properties (Bapat and Raghavan 1997):

(i) $\lim_{k\to\infty}\mathbf{P}^k = \mathscr{P}$ and $\mathbf{P}\mathscr{P} = \mathscr{P}\mathbf{P} = \mathscr{P} = \mathscr{P}^2$

(ii) The matrix $\mathscr{P}$ has the following structure:

$$\mathscr{P} = \begin{bmatrix}\wp^T\\ \cdots\\ \wp^T\end{bmatrix} \quad \text{where } \wp^T\mathbf{P} = \wp^T$$

implying that $\wp$ is the left eigenvector of **P** corresponding to its unique unity eigenvalue

(iii) Upon ℓ_1-normalization, $\wp$ becomes the state probability vector of the stationary Markov chain associated with the stochastic primitive matrix **P**.

(iv) The spectral radius of the matrix $(\mathbf{P}-\mathscr{P})$ is less than unity, i.e., the eigenvalues of $(\mathbf{P}-\mathscr{P})$ are located within the unit radius circle with center at the origin.

For a primitive stochastic matrix, the expression for $\boldsymbol{\vartheta}^0|_p$ in equation (19) of Proposition 6 is simplified as presented in the following proposition.

Proposition 7: *For a primitive stochastic matrix* **P**, *the generalized measure is expressed as*

$$\boldsymbol{\vartheta}^0|_p \equiv \lim_{\theta\to 0^+}\boldsymbol{\vartheta}^\theta|_p = \frac{\wp^T\boldsymbol{\chi}}{\|\wp\|_p}\begin{Bmatrix}1\\ \vdots\\ 1\end{Bmatrix}, \tag{20}$$

where $\wp^T\mathbf{P} = \wp^T$.

Proof: From the properties (i) and (ii) of primitive matrices, it follows that

$$\mathscr{P}_{j\circ} = \wp^T \;\forall j\in\{1,\ldots,n\}, \tag{21}$$

where $\wp^T$ is the state probability vector of the associated Markov chain. Then, the proof follows from Proposition 6. □

3.2 *Physical interpretation of the* $\boldsymbol{\vartheta}^0|_p$ *measures*

All entries of the $\boldsymbol{\vartheta}^0|_p$ vector in equation (19) are identical for a primitive stochastic matrix and hence a single entry can be taken as a scalar measure, $\vartheta^0|_p \equiv \wp^T\chi/\|\wp\|_p$, of the regular language of the underlying automaton. For all $p\in[1,\infty]$, the measure $\vartheta^0|_p$ represents the long-range behaviour of the plant dynamics in terms of the (assigned) terminating characteristics and the stationary state probability vector of the finite Markov chain model. However, the measures for different values of p are not equivalent in the sense that a control policy optimizing $\vartheta^0|_p$ does not necessarily coincide with one that optimizes $\vartheta^0|_q$ for $p\neq q$. For example, a control policy that maximizes $\vartheta^0|_1$ selectively disables controllable events such that $\wp^T\boldsymbol{\chi}$ is maximized; and a control policy that maximizes $\vartheta^0|_2$ chooses an automaton configuration to make the stationary state probability vector $\wp$ closest to the terminal characteristic vector $\boldsymbol{\chi}$ in the Euclidean sense. For physical understanding and visualization, let $\mathscr{S}$ be a bounded submanifold of $\mathbb{R}^n$ such that

$$\forall\, p = \{p_1,\ldots,p_n\} \in \mathscr{S} \quad \text{with} \begin{cases} p_i \geq 0\\ \sum_{i=0}^n p_i = \|p\|_{l_1} = 1.\end{cases} \tag{22}$$

Then, for any n-state automaton, the stationary state probability vector is $\wp\in\mathscr{S}$. Figure 2 illustrates $\mathscr{S}$ for

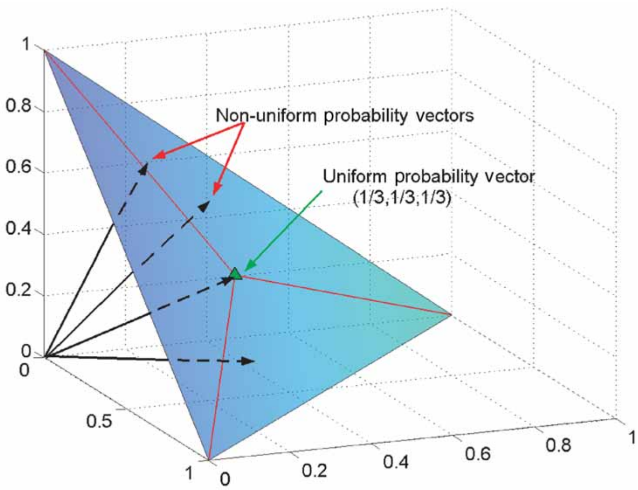


Figure 2. Representation of the $\mathscr{S}$-plane for three states.

$n=3$, where the central point $\wp_c$ denotes the uniform probability vector $[1/n, \ldots, 1/n]$, which is interpreted to have maximum entropy $\log_2 n$ in the Shannon sense (Cover and Thomas 1991). Moving away from $\wp_c$ on the $\mathscr{S}$-plane, the distribution becomes non-uniform, i.e., the Shannon entropy $S \equiv -\sum_{k=1}^{n}\left(p_k \log_2 p_k\right)$ decreases toward zero.

In view of the above discussion and Lemma 4, a supervisory control policy can be constructed by optimizing $\vartheta^0|_p$ in the sense of equation (19) for a specified $p \in [1, \infty]$ to obtain a stationary state probability vector $\wp$. For example, if $p=1$, then the optimization algorithm attempts to choose $\wp$ as a unit vector in the direction of one of the axes of $\mathbb{R}^n$ for which the $\boldsymbol{\chi}$-vector has the largest element; if this is the case, then Shannon entropy $S=0$. If $p=2$, then the optimization algorithm attempts to choose $\wp$ as the point of intersection of the $\boldsymbol{\chi}$ vector with the $\mathscr{S}$-plane; in this case, the Shannon entropy is $S>0$ unless $\boldsymbol{\chi}$ is coincident with one of the axes of $\mathbb{R}^n$. For $p>2$, the algorithm attempts to choose $\wp$ closer to the central point $\wp_c$ more and more aggressively as p increases toward infinity, for which the Shannon entropy S increases toward its maximum value $\log_2 n$.

Three measures are considered to be significant; $\vartheta^0|_1$ and $\vartheta^0|_\infty$ optimal policies are useful to obtain low and high entropy (thermodynamically stable) distributions, respectively, and $\vartheta^0|_2$ optimality is useful when the problem definition requires achieving a target distribution over the plant states as closely as the controllability criteria would allow. An example is given in § 5.

The following lemma is useful for interpretation of the $\boldsymbol{\vartheta}^0|_p$ measures.

Lemma 4: *Let ν be a n-dimensional vector with $\nu_i \in [0,1]$ and $\sum_i \nu_i = 1$. Then we have*

$$\text{(i)} \quad \|\nu\|_p \in \left[n^{(1-p)/p}, 1\right] \quad \forall p \in [1, \infty] \tag{23}$$

$$\text{(ii)} \quad \|\nu\|_p \le \|\nu\|_q \quad \forall p > q \text{ with } p, q \in [1, \infty] \tag{24}$$

Proof: For Assertion (i),

$$\nu_i \in [0,1] \;\Rightarrow\; \nu_i^p \le \nu_i \;\Rightarrow\; \sum_i \nu_i^p \le \sum_i \nu_i \;\Rightarrow\; \|\nu\|_p \le 1.$$

The result follows by noting that the smallest value is attained when all ν_i are equal i.e. $\nu_i = 1/n\ \forall i$. Assertion (ii) follows by noting that $\nu_i^p \le \nu_i^q$ if $p>q$. □

4. Shaped measures

The measures defined in the previous sections put equal importance to all traces in the generated language of an automaton, including traces of unbounded lengths. This section investigates formal measures that generalize the process of assigning importance or weight to a trace as a function of its length. For a given automaton G_i, a partition of the generated language $L(G_i)$ is obtained as:

$$L(G_i) = \bigcup_{i=0}^{\infty} \mathscr{L}_i^r \quad \text{where} \quad \mathscr{L}_i^r = \left\{\omega \in L(q_i, q_j) : |\omega| = r\right\}. \tag{25}$$

(Note that $\mathscr{L}_i^r \bigcap_{r \ne s} \mathscr{L}_i^s = \emptyset$.) From σ-additivity of the language measure (Ray 2005), the following notion of language measure is introduced.

Definition 10: For a given starting state q_i and the parameter $\theta \in (0,1)$, the shaped measure of the language $L(G_i)$ is defined as

$$\nu(L(G_i)) = \sum_{r=0}^{\infty} \mu_i^\theta(\mathscr{L}_i^r). \tag{26}$$

The above definition, as observed before, fails to exist as $\theta \to 0^+$. The singularity at $\theta=0$ is alleviated by a shaping sequence that provides appropriate weights on the individual terms of the infinite sum in equation (26). The next proposition establishes that every ℓ_1-sequence qualifies as a shaping sequence.

Proposition 8: *For a real ℓ_1-sequence $\Gamma = \{\gamma_i\}$ with $\gamma_i \in [0, \infty)$,*

$$\text{(i)} \sum_{i=0}^{\infty} \mu_i^{\theta}(\mathscr{L}^i)\gamma_i < \infty \quad \forall\, \theta \in (0,1) \tag{27}$$

$$\text{(ii)} \lim_{\theta \to 0^+} \sum_{i=0}^{\infty} \mu_i^{\theta}(\mathscr{L}^i)\, \gamma_i < \infty. \tag{28}$$

Proof: The proof of the proposition requires the following lemma. □

Lemma 5: *The following expression holds for the θ-parameterized shaped measure*

$$\boldsymbol{\mu}^{\theta}(\mathscr{L}_i^r) = (1-\theta)^r \mathbf{P}^r \boldsymbol{\chi}. \tag{29}$$

Proof: From Definition 7, we have

$$\begin{aligned}
\mu_i^{\theta}(\mathscr{L}_i^r) &= \sum_{j=1}^{n} \sum_{\substack{\omega \in \\ L(q_i,q_j)\cap \mathscr{L}_i^r}} \tilde{\pi}(q_i,\omega)\chi_j \\
&= \sum_{j=1}^{n} \left\{ \sum_{i_1} \cdots \sum_{i_r} \pi_{ii_1} \cdots \pi_{i_r j} \right\} \chi_j \\
&= \sum_j \Pi_{ij}^r \chi_j = \sum_j (1-\theta)^r \mathbf{P}_{ij}^r \chi_j
\end{aligned}$$

Using Lemma 5 and noting that for all $\theta \in [0,1)$, it follows that

$$\|(1-\theta)^r \mathbf{P}^r \boldsymbol{\chi}\|_\infty \le |1-\theta|^r \; \|\mathbf{P}\|_\infty^r \; \|\boldsymbol{\chi}\|_\infty \le 1, \tag{30}$$

where $\| \bullet \|_\infty$ is the induced sup-norm of $\bullet$. □

The shaped measure is now formally defined based on Proposition 33 by setting the parameter θ to 0.

Definition 11: Let $\Gamma = \{\gamma_i\}$ be a ℓ_1-sequence of non-negative real numbers (called the shaping sequence in the sequel). The shaped measure τ_i^Γ of a trace set $\{s\} \subseteq L(q_i, q_j) \subseteq L(G_i)$ with $s = k \in \mathbb{N} \cup \{0\}$ relative to Γ is defined as

$$\tau_i^\Gamma(\{s\}) \equiv \mu^\theta(\{s\})_k = \tilde{\pi}(s, q_i)\chi(q_j)_k \quad \forall s \in L(q_i, q_j). \tag{31}$$

The shaped measure of $L(q_i, q_j)$ is defined as:

$$\tau_i^\Gamma\big(L(q_i,q_j)\big) \equiv \sum_{r=0}^{\infty} \sum_{\substack{\omega \in \\ L(q_i,q_j)\cap \mathscr{L}_i^r}} \tau_i^\Gamma(\{s\}). \tag{32}$$

The shaped measure of a *DFSA* G_i, relative to the sequence Γ and initialized at the state $q_i \in Q$, is denoted as: $\tau_i^\Gamma \equiv \tau_i^\Gamma(L(G_i)) = \sum_j \tau_i^\Gamma(L(q_i, q_j))$.

The shaped measure vector, relative to the sequence Γ, is denoted as: $\tau^\Gamma \equiv [\tau_1^\Gamma, \ldots, \tau_n^\Gamma]$.

Remark 2: If the short-term behaviour of the discrete-event system is of interest, then all but finitely many elements of the shaping sequence $\Gamma = \{\gamma_i\}$ in Definition 11 could be restricted to be zeros. Then, there exists $r^\star \in \mathbb{N}$ such that $\mathscr{L}_i^r = \emptyset \; \forall r \ge r^\star$, i.e., the generated language has only bounded length traces.

4.1 *Relation between $\tau^{\Gamma(p)}$ and $\boldsymbol{\vartheta}^0|_p$ measures*

In spite of a different construction, shaped measures are related to the generalized measure defined in §3. Specifically, there exist sequences of shaped measures that converge to $\boldsymbol{\vartheta}^0|_p$.

Remark 3: Let $p \in [1, \infty]$ and let $\Gamma_k(p), k \in \mathbb{N}$ be a sequence of non-negative real numbers, whose all elements, except the kth one, are zeroes and the kth element is $\|\wp\|_p$. Let $\Gamma(p) \equiv \lim_{k\to\infty} \Gamma_k(p)$. Then, it follows from Proposition 6 or Proposition 7 that there exists $\Gamma(p)$ such that $\tau^{\Gamma(p)} = \boldsymbol{\vartheta}^0|_p \; \forall p \in [1, \infty]$.

4.2 *Physical interpretation of shaped measures*

A shaping sequence Γ specifies length-based relative importance of traces in the generated language. Intuitively, one is rarely interested in all traces generated by an automaton. More often than not, either short traces or very long traces (specifically of unbounded length) are important. The first case is handled by shaping sequences with finitely many non-zero terms and the latter, shown in Remark 3 is viewed as a limit of the shaped measures. However, shaping sequences can be more complicated; the only requirement is that the sequence be in ℓ_1 (see Proposition 8). In this context, Remark 3 implies that $\vartheta^0|_1$ addresses the long-term behaviour of the discrete-event system based on the traces of unbounded length with no importance to finite traces. This follows from the fact that, for $p = 1$ and all elements of the sequence Γ_k are zeros with the exception of the k^{th} element being equal to 1.

5. An illustrative example

Figure 3 shows the finite-state automaton model of the plant, where the state set $Q = \{q_1, \ldots, q_9\}$ and the event alphabet $\Sigma = \{\sigma_r, \sigma_l, \sigma_f, \sigma_b, \sigma_{fl}, \sigma_{rf}, \sigma_{rb}, \sigma_{lb}, \omega_1\}$. The transitions, shown by dashed lines, are controllable and those, shown by solid lines, are uncontrollable. The state transition matrix $\mathbf{P}$ is given in table 1. The stochastic matrix $\mathbf{P}$ is primitive because $\mathbf{P}^2$ is a positive matrix. The stationary state probability vector of $\mathbf{P}$ and the

χ-vector for the *DFSA* are

$$\wp^T = \begin{bmatrix} 0.234 & 0.041 & 0.065 & 0.084 & 0.095 & 0.016 & 0.153 & 0.147 & 0.076 \end{bmatrix}$$
$$\chi = \begin{bmatrix} 0.66 & -0.42 & -0.97 & 0.52 & -0.49 & -0.57 & 0.57 & -0.09 & 0.43 \end{bmatrix}^T.$$

The scalar measures $\vartheta^0|_p$ and $\|\wp\|_p$ for the primitive matrix in table 1 are plotted for different values of p in figure 4. A shaping sequence Γ evaluates the short-term behaviour based on traces of length less than 60 and the resulting measure vector is

$$\tau^\Gamma = \begin{bmatrix} 0.276 & -0.252 & 0.627 & -0.375 & 0.256 & -0.055 & -0.059 & -0.103 & 0.475 \end{bmatrix}^T.$$

Supervisory control policies have been computed based on $\vartheta^0|_p$, $\|\wp\|_p$ and τ^Γ by optimizing the respective scalar measures via a standard search algorithm. Different values of $p = 1, 4,$ and ∞ are chosen to illustrate the fact that they result in different optimal control policies. The choices of $p=1$ and $p=\infty$ are made as they are familiar norms used in engineering analysis; and the choice of $p=4$ is made because it effects are intermediate between $p=1$ and $p=\infty$ and are different from those of the Euclidean norm $p=2$ (see §3). For these cases, the results of improved performance under optimal supervision are summarized below.

For $p=1$, $\vartheta^0|_1$ is increased from 0.12 to 0.35.
For $p=4$, $\vartheta^0|_4$ is increased from 0.48 to 1.03.
For $p=\infty$, $\vartheta^0|_\infty$ is increased from 0.52 to 1.3 and τ^Γ is increased elementwise from

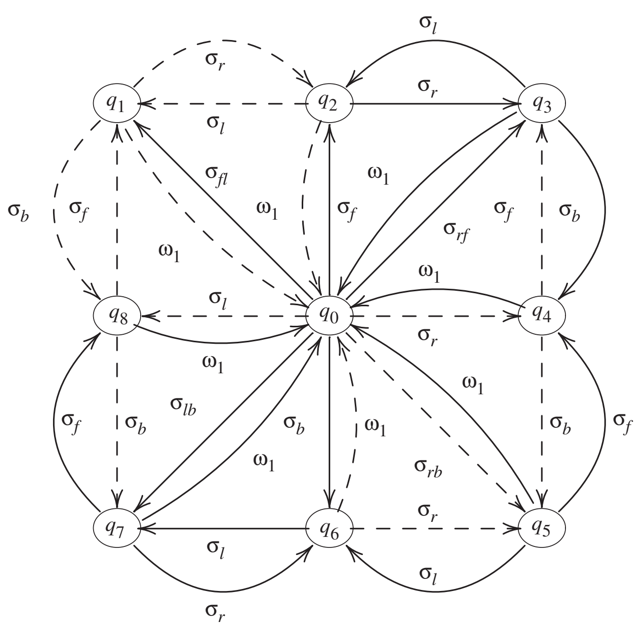


Figure 3. Plant automaton model.

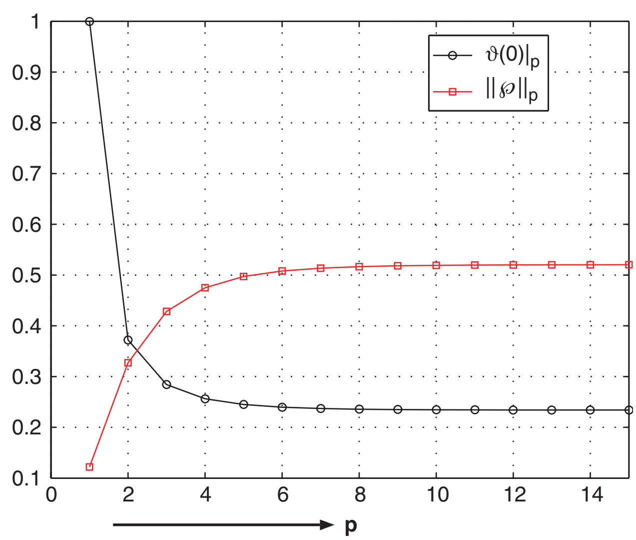


Figure 4. Profiles of $\vartheta^0|_p$ and $\|\wp\|_p$ with p. Note the stabilizing feature of each plot for increasing p.

Table 1. State transition matrix **P** of the plant automaton model.

0	0.015	0.102	0.041	0.120	0.048	0.300	0.139	0.139
0.372	0	0.131	0	0	0	0	0	0
0.130	0.319	0	0.551	0	0	0	0	0
0.087	0	0.424	0	0.489	0	0	0	0
0.351	0	0	0.411	0	0.238	0	0	0
0.337	0	0	0	0.240	0	0.423	0	0
0.069	0	0	0	0	0.470	0	0.460	0.460
0.738	0	0	0	0	0	0.259	0	0
0.199	0.218	0	0	0	0	0	0.583	0.583

Table 2 Optimal decision for disabling of controllable events.

Controllable events	$\vartheta^0\|_1$	$\vartheta^0\|_4$	$\vartheta^0\|_\infty$	τ^Γ
$q_1 \overset{\sigma_r}{\to} q_5$	×	×	×	×
$q_1 \overset{\sigma_{rb}}{\to} q_6$	×	×	×	×
$q_1 \overset{\sigma_l}{\to} q_9$	✓	✓	✓	✓
$q_2 \overset{\omega_1}{\to} q_1$	✓	✓	✓	✓
$q_2 \overset{\sigma_r}{\to} q_3$	×	×	×	×
$q_2 \overset{\sigma_b}{\to} q_9$	✓	×	✓	×
$q_3 \overset{\omega_1}{\to} q_1$	✓	✓	✓	✓
$q_5 \overset{\sigma_f}{\to} q_4$	✓	✓	✓	✓
$q_5 \overset{\sigma_b}{\to} q_6$	✓	✓	✓	✓
$q_7 \overset{\omega_1}{\to} q_1$	×	×	×	✓
$q_7 \overset{\sigma_r}{\to} q_6$	×	×	×	×
$q_9 \overset{\sigma_b}{\to} q_8$	×	×	×	×
$q_9 \overset{\sigma_f}{\to} q_2$	×	×	✓	×

$[1.2\ \ 1.5\ \ 0.9\ \ 0.3\ \ 1.1\ \ 1.4\ \ 1.2\ \ 1.7\ \ 1.4]^T$ to $[3.4\ \ 3.8\ \ 2.5\ \ 1.7\ \ 2.6\ \ 3.3\ \ 3.5\ \ 3.7\ \ 4.0]^T$.

Table 2 enumerates the optimal decision sets for disabling of controllable events obtained in the above four cases, where × and ✓ indicate disabled controllable events and enabled controllable events, respectively. The decisions are made from the stationary state probability distributions achieved from the optimal policies shown in figure 5. It is seen that the $\vartheta^0|_1$-optimal policy achieves both maximum and minimum probability values in states 9 and 6, respectively. This shows that $\vartheta^0|_1$-optimal policy does indeed produce relatively less uniform distribution in comparison to the $\vartheta^0|_\infty$-optimal policy, as stated in § 3. It is also noted that the τ^Γ-optimal policy achieves the least uniform distribution.

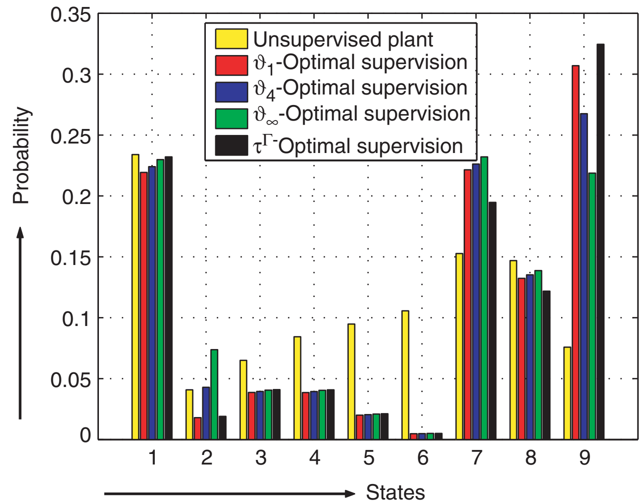


Figure 5. Stable distributions for computed optimal policies.

6. Summary, conclusions, and future work

This paper formulates and validates a concept of generalization of signed real measure of regular languages, which also leads to renormalization (Chattopadhyay and Ray 2006) of the normalized measure and eliminates the need for a user-selectable parameter in the original concept of language measure Ray (2005). These generalizations are achieved through a trace measure that is characterized by both initial and terminal states as well as the length of the trace and the choice of a vector norm for renormalization. The generalized measures with different norms are not equivalent in the sense that the respective optimal control policies with these measures as the performance cost functionals are different. These concepts are illustrated with simple examples for quantitative analysis and synthesis of discrete-event supervisory control systems. It is envisioned that optimal supervisory decision & control of discrete-event systems (Sengupta and Lafortune 1998, Ray *et al*. 2004) can be enhanced through appropriate selection of a language measure to enhance the objectives at hand. In this context, future research is recommended in the following areas:

- Generalization of the language-measure-based optimal control algorithms (Ray *et al*. 2004) for sub-stochastic transition matrices to the stochastic case. A potential application is to compute a sufficiently small termination probability θ such that, as $\theta \to 0^+$, the optimal control policies approach the true situation for the non-terminating plant.
- Extension the concept of (regular) language measure for (non-regular) languages higher up in the Chomsky Hierarchy such as context-free and context -sensitive languages. A first attempt to extend the concept of the language measure to linear grammars was reported in (Ray *et al*. 2004). Further investigations in this direction is required for extension of the concept to more complex models.
- Applications of language measure in anomaly detection, model identification and order reduction, and construction of interfaces between continuously varying and discrete-event spaces.

Acknowledgements

This work has been supported in part by the U.S. Army Research laboratory and the U.S. Army Research Office

under Grant Nos. DAAD19-01-1-0646 and W911NF-06-1-0469.

References

R. Bapat and T. Raghavan, *Nonnegative Matrices and Applications*, Cambridge, UK: Cambridge University Press, 1997.

I. Chattopadhyay and A. Ray, "A complex measure for linear grammars", *Demonstratio Mathematica*, 38, pp. 761–775, 2005.

I. Chattopadhyay and A. Ray, "Renormalized measure of regular languages", in *Proceedings of WODES 2006*, submitted for review, (Ann Arbor, MI), July 2006.

T. Cover and J. Thomas, *Elements of Information Theory*, New York: John Wiley, 1991, pp. 18–19.

V. Garg, "An algebraic approach to modeling probabilistic discrete event systems", in *Proceedings of 1992 IEEE Conference on Decision and Control*, Tucson, AZ, December 1992a, pp. 2348–2353.

V. Garg, "Probabilistic lnaguages for modeling of DEDs", in *Proceedings of 1992 IEEE Conference on In formation and Sciences*, Princeton, NJ, March 1992, pp. 198–203.

J. Hopcroft, R. Motwani and J. Ullman, *Introduction to Automata Theory, Languages, and Computation*, 2nd ed., Boston, MA, USA: Addison-Wesley, 2001.

A. Naylor and G. Sell, *Linear Operator Theory in Engineering and Science*, New York, USA: Springer-Verlag, 1982.

P. Ramadge and W. Wonham, "Supervisory control of a class of discrete event processes", *SIAM J. Cont. Optim.*, 25, pp. 206–230, 1987.

A. Ray, "Signed real measure of regular languages for discrete-event supervisory control", *Int. J. Cont.*, 78, pp. 949–967, 2005.

A. Ray, J. Fu and C. Lagoa, "Optimal supervisory control of finite state automata", *Int. J. Cont.*, 77, pp. 1083–1100, 2004.

A. Ray, V. Phoha and S. Phoha, *Quantitative Measure for Discrete Event Supervisory Control*, New York: Springer, 2005.

R. Sengupta and S. Lafortune, "An optimal control theory for discrete event systems", *SIAM J. Cont. Optim.*, 36, pp. 488–541, 1998.